



Titre: Segmentation basée sur le mouvement dans le cadre d'une caméra
Title: omnidirectionnelle non-statique

Auteur: Jean Herbière
Author:

Date: 2006

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Herbière, J. (2006). Segmentation basée sur le mouvement dans le cadre d'une
Citation: caméra omnidirectionnelle non-statique [Mémoire de maîtrise, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/7717/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7717/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

SEGMENTATION BASÉE SUR LE MOUVEMENT DANS LE CADRE D'UNE
CAMÉRA OMNIDIRECTIONNELLE NON-STATIQUE

JEAN HERBIERE

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GROUPE DE RECHERCHE EN PERCEPTION ET ROBOTIQUE)

AVRIL 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-17946-8

Our file Notre référence

ISBN: 978-0-494-17946-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

SEGMENTATION BASÉE SUR LE MOUVEMENT DANS LE CADRE D'UNE
CAMÉRA OMNIDIRECTIONNELLE NON-STATIQUE

présenté par: HERBIERE Jean

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. GOURDEAU Richard, Ph.D., président

M. COHEN Paul, Ph.D., membre et directeur de recherche

M. BILODEAU Guillaume-Alexandre, Ph.D., membre

À ma mère.

REMERCIEMENTS

Tout d'abord, je voudrais remercier M. Paul Cohen de m'avoir accueilli dans le Groupe de Recherche en Perception et Robotique et de m'avoir encadré tout au long de ma maîtrise.

Je voudrais exprimer ma reconnaissance envers tous les membres du GRPR que j'ai côtoyé durant mes recherches. En particulier je tiens à remercier deux personnes qui ont été à mes côtés à toutes les étapes de mon projet. Tout d'abord Vincent Paquin, assistant de recherche au laboratoire, dont le soutien a été autant théorique, technique que moral dans les moments où mes calculs donnaient les pires résultats de l'histoire de la Vision par Ordinateur. Ensuite, j'aimerais saluer la patience et la pédagogie dont Vincent Zalzal a fait preuve devant le montagne de difficultés que constituait la programmation pour moi. C'est à présent pour moi une colline, mais riche de découvertes et d'expériences à venir.

Enfin, je témoigne ma reconnaissance à ma famille proche qui a su m'encourager dans les moments d'épreuves. Je peux affirmer que je sors de cette maîtrise avec des connaissances élargies, et une expérience du travail en laboratoire de recherche qui, je suis sûr, me suivra toute ma vie.

RÉSUMÉ

La surveillance est un des domaines d'application majeurs de l'analyse d'image et de la vision par ordinateur. Les outils développés sont aussi bien matériels, comme de nouveaux types de caméras, que logiciels, avec de nouveaux programmes de traitement d'image. La détection de mouvement dans une image reste un sujet qui est sans cesse à améliorer. La grande difficulté de ce problème concerne la différenciation dans les images capturées par un système visuel en mouvement entre, l'effet produit par un objet en mouvement indépendant et l'effet induit par le déplacement relatif entre les parties de la scène et le système de vision. Le flux optique reste l'un des outils les plus fréquemment utilisés pour résoudre ce problème connu sous le nom de segmentation de mouvement. Par ailleurs, nous devons également utiliser au mieux les nouveaux systèmes visuels pour permettre cette détection.

Les caméras omnidirectionnelles à miroirs coniques sont de plus en plus utilisées en vision. Elles fournissent un large champ de vision avec des contraintes matérielles minimales. Toutefois, elles nécessitent de nouvelles études théoriques afin de rendre compte de leur géométrie projective atypique. Afin d'adapter les outils de détection déjà développés pour des caméras normales, dites «en perspective», aux caméras omnidirectionnelles, il est nécessaire de prendre en compte les déformations importantes présentes dans l'image ainsi que la variabilité de la résolution au sein des images omnidirectionnelles, lesquelles diffèrent d'un type de miroir à un autre.

Le premier objectif du projet a été de connaître la géométrie projective de ces caméras pour pouvoir adapter au cas omnidirectionnel les outils déjà développés pour les caméras traditionnelles. Comme le flux optique constitue une des données essentielles lors d'une détection de mouvement, nous avons commencé par exprimer le flux théorique dans une image omnidirectionnelle pour un nombre fini de mouvements, puis nous avons adapté une des techniques de calcul de flux optique

pour qu'elle prenne en compte la géométrie projective de ces caméras. La précision des calculs s'est trouvée améliorée par notre méthode.

Nous avons par la suite développé la théorie nécessaire à une segmentation de mouvement pour une caméra omnidirectionnelle en déplacement. Elle a permis une détection de mouvement efficace pour différents types de mouvements. Elle fonctionne en temps réel et permet une détection d'un grand nombre de zones en mouvement dans l'image, et a été testée dans le cadre de séquences synthétiques aussi bien que réelles.

Le grand champ de vision est un atout mais la partie centrale de l'image est occupée par la réflexion de la caméra. L'originalité de notre approche consiste à coller aux dos de la caméra omnidirectionnelle une autre caméra pour permettre d'obtenir de la caméra omnidirectionnelle un grand angle de vision et de la seconde caméra une précision et une résolution plus importante dans la direction observée. Notre système est posé sur un double moteur, pour le tangage et le lacet, et permet ainsi de fournir des mouvements calibrés en amplitude et en vitesse pour notre détection. Certaines limitations sont néanmoins présentes dans notre projet : (1) les nouveaux algorithmes de calcul de flux optique adaptés ne sont pas encore optimisés en temps et devraient faire l'objet de développement ultérieurs et l'adaptation pourrait concerner plus de techniques de calcul. (2) le nombre de mouvements pour lesquels la détection est possible est encore limité et l'élargissement à tous les mouvements possibles est encore à développer.

Finalement, les contributions scientifiques sont tout d'abord théoriques avec le développement d'équations sur les caméras omnidirectionnelles nécessaires à l'expression du flux théorique dans une image omnidirectionnelle et à une adaptation de son calcul. Elles relèvent ensuite codage des projections et la mise en place d'algorithmes de détection dans le cas omnidirectionnel, et finalement, de la construction d'un système de vision original réunissant toutes les composantes précédentes.

ABSTRACT

Surveillance is one of the major application of image analysis and computer vision. The developed tools are material, like new types of cameras, and software, with new programmes of image processing. The detection of movement in an image, called motion segmentation is unceasingly to improve. If a visual system is moving, every pixel may undergo apparent motion, and the resulting apparent motion field, the optical flow, can be complex. Then, detection of moving objects requires that we are able to make the distinction between apparent motion that is only due to the movement of the camera, and apparent motion that arises from the relative movement of an independently moving object. In addition, we must also use the new visual systems as well as possible to allow this detection.

Omnidirectional cameras with conical mirrors are used more and more in vision. They provide a broad field of vision with tiny material constraints. However, they require new theoretical studies in order to give an account of their atypical projective geometry. In order to adapt the tools for detection already developed for perspective cameras with the omnidirectional cameras, it is necessary to take into account the significant deformations present in the image as well as the variability of the resolution within the omnidirectional images, which differ from a type of mirror to another.

The first objective of the project was to know the projective geometry of these cameras to be able to adapt to the omnidirectional case the tools already developed for the traditional cameras. As optical flow constitutes one of the essential data during a detection of movement, we started by expressing theoretical flow in an omnidirectional image for a finished number of movements, then we adapted one of the techniques of optical calculation of flow so that it takes into account the projective geometry of these cameras. The precision of calculations was improved by our method.

Thereafter, we developed the theory necessary to a motion detection for an omnidirectional moving camera. It allowed an effective detection of movement and that with various types of movements. It functions in real time and allows a detection of a great number of zones moving in the image. This protocol was tested within the framework of synthetic sequences as well as real.

The large field of vision is an asset but the central part of the image is occupied by the reflection of the camera. The originality of our approach consists in sticking to the backs of the omnidirectional camera another camera to make it possible to obtain from the omnidirectional camera a great angle of vision, and from the second camera, a precision and a more significant resolution in the direction observed. Our system is posed on a double engine, for the pitching and the lace, and thus makes it possible to provide movements gauged in amplitude and of speed for our detection. Limitations are nevertheless present in the our projet :(1) new calculation algorithms of adapted flows optical is not optimized yet in time and should be the subject later of development and the adaptation could relate to more techniques of calculation. (2) the number of movements for which detection is possible is still limited and widening with all the possible movements is still to develop.

Finally, the scientific contributions are first of all theoretical, with the development of equations on the omnidirectional cameras necessary to the expression of theoretical flow in an omnidirectional image and for an adaptation of its calculation. They concern then the implementation of the various projections and the algorithms of detection in the omnidirectional case. Finally, the construction of an original system of vision allowed us to join together all the preceding components .

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES FIGURES	xv
LISTE DES NOTATIONS ET DES SYMBOLES	xxiv
LISTE DES TABLEAUX	xxv
LISTE DES ANNEXES	xxvii
INTRODUCTION	1
CHAPITRE 1 POSITIONNEMENT DU SUJET	4
1.1 Revue Bibliographique	4
1.1.1 Systèmes catadioptriques	4
1.1.1.1 Modèles mathématiques	4
1.1.1.2 Calibration des caméras omnidirectionnelles	7
1.1.1.3 Calcul du flux optique dans les images omnidirectionnelles	11
1.1.1.4 Calcul du mouvement propre avec une caméra omnidirectionnelle	12

1.1.2	Segmentation basée sur le mouvement dans le cadre d'une caméra omnidirectionnelle	14
1.2	Description du projet	17
1.2.1	Problématiques	18
1.2.2	Objectifs	21
1.2.3	Considérations techniques	22
1.2.3.1	Partie montage	22
1.2.3.2	Partie logicielle	23
CHAPITRE 2	BASE DE LA THÉORIE DES CAMÉRAS OMNIDIRECTIONNELLES	24
2.1	Point de vue unique et forme des miroirs	24
2.1.1	Calcul de l'équation de contrainte avec une caméra en perspective	25
2.1.2	Solution générale de l'équation de contrainte	26
2.1.2.1	Miroir plan	28
2.1.2.2	Miroir en forme de cône	28
2.1.2.3	Miroir Sphérique	29
2.1.2.4	Miroir ellipsoïdal	29
2.1.2.5	Miroir hyperboloïdal	30
2.1.3	Calcul de l'équation de contrainte dans le cas d'une lentille orthographique	30
2.2	Cas du miroir hyperboloïdal	32
2.3	Théorie unificatrice	35
2.3.1	Premières Équations	35
2.3.2	Projection inverse sur la sphère	37
2.4	Projection de droites sur le plan image	40
2.4.1	Projection des droites sur la sphère	40

2.4.2	Projection des grands cercles sur le plan image	41
2.4.3	Cas Particulier : droites parallèles au plan image	42
CHAPITRE 3 CALCUL DU FLUX OPTIQUE DANS DES IMAGES OM-		
	NIDIRECTIONNELLES	46
3.1	Introduction au flux optique	46
3.1.1	Premiers calculs	46
3.1.2	Méthode différentielle avec hypothèse de constance locale . .	48
3.2	Opérateurs appliqués sur des images omnidirectionnelles	51
3.2.1	Problèmes avec les opérateurs habituels	51
3.2.2	Transposer le calcul sur la sphère	52
3.2.3	Opérateurs à adapter	54
3.2.4	Adaptation des opérateurs	57
3.2.4.1	Projection inverse sur un plan $Z = Z_0$	60
3.2.4.2	Voisinage	61
3.2.4.3	Opérateurs de dérivation	64
3.3	Méthode d'analyse de l'algorithme de flux optique	66
3.3.1	Choix d'une variable pour analyser l'effet des nouveaux opérateurs sur le calcul du flux optique	66
3.3.2	Flux optiques calculés	67
3.4	Flux optique théorique	69
3.4.1	Rotation autour de l'axe de la caméra	69
3.4.2	Translation dans la direction de l'axe de la caméra	70
3.4.3	Translation perpendiculaire à l'axe de la caméra	71
3.5	Système utilisé	75
3.6	Validité du calcul de flux optique	77
3.6.1	Figures simples	79
3.6.1.1	Déplacement de la forme : premier exemple	80

3.6.1.2	Changement de position dans l'image et variation d'amplitude	82
3.6.2	Validation sur des images synthétiques	87
3.6.3	Validation sur des images réelles	99
3.7	Comparaison avec les opérateurs homogènes	104
3.8	Observation par couronne	109
3.8.1	Séquences d'images synthétiques	109
3.8.2	Séquences d'images réelles	115
3.8.3	Conclusion sur la comparaison entre algorithmes	119
3.9	Comparaison avec d'autres algorithmes	120
3.10	Conclusion sur l'adaptation des opérateurs pour le calcul de flux optique dans les images omnidirectionnelles	123
CHAPITRE 4 SYSTÈME VISUEL ET PROTOCOLE DE DÉTECTION DE MOUVEMENT 125		
4.1	Système visuel	125
4.1.1	Architecture matérielle	125
4.1.2	Architecture logicielle	127
4.2	Protocole de détection	127
4.2.1	Modèle de mouvement pour une caméra omnidirectionnelle .	128
4.2.2	Relier le flux optique au modèle de mouvement	130
4.2.3	Schéma du protocole	131
4.2.4	Particularités du montage et changements de repère	132
4.2.5	Système d'équations dans le cadre des moteurs PAN-TILT .	136
4.2.5.1	Mouvement PAN	137
4.2.5.2	Mouvement TILT	137
4.3	Seuil de détection	137
4.4	Création de zones en mouvement indépendant et suivi	140

4.5	Résultats	141
4.5.1	Analyse de séquences d'images synthétiques	141
4.5.1.1	Création de la scène	141
4.5.1.2	Analyse des mouvements dans la scène synthétique	142
4.5.2	Séquences réelles	147
4.5.2.1	Personne en déplacement	148
4.5.2.2	Plate-forme en déplacement	153
4.6	Améliorations futures	155
4.7	Relevé d'informations à plus haute résolution	156
4.8	Conclusion sur le protocole de détection de mouvement	159
CONCLUSION		161
RÉFÉRENCES		164
ANNEXES		170

LISTE DES FIGURES

FIG. 1	Le champ de vision est de 360° autour de l'axe de la caméra.	2
FIG. 1.1	Dans le cas d'un miroir hyperboloïdal : les rayons se dirigent vers le point de vue effectif, ici le foyer de l'hyperboloïde, se réfléchissent sur le miroir et se dirigent vers le point de vue réel de la caméra, situé au second foyer de l'hyperboloïde, et touchent le plan image.	6
FIG. 1.2	Illustration des paramètres de la matrice de calibration. . .	8
FIG. 1.3	Translation perpendiculaire à l'axe de la caméra : le foyer d'expansion Q' et le foyer de contraction P' sont présents. .	13
FIG. 1.4	Deux types de projection sur le plan image.	19
FIG. 1.5	Types de mouvement considérés dans la littérature.	20
FIG. 1.6	Système utilisé : deux caméras et un miroir hyperboloïdal posés sur un double moteur	23
FIG. 2.1	Géométrie utilisée pour déterminer les formes de miroirs possibles dans le cadre d'un point de vue unique.	25
FIG. 2.2	Géométrie utilisée pour déterminer les formes de miroirs possibles dans le cadre d'un point de vue unique et d'une lentille orthographique.	31
FIG. 2.3	Caméra omnidirectionnelle avec un miroir de forme hyperboloïdale	32
FIG. 2.4	Projections successives	35
FIG. 2.5	Pour des points situés derrière le foyer de l'hyperbole sur l'axe des Z , leur image est à l'extérieur de l'horizon fronto-parallèle. Pour des points devant le foyer, l'image est à l'intérieur de l'horizon.	38
FIG. 2.6	Projections de droites sur le plan image via les deux projections	41

FIG. 2.7	Projection d'une droite sur le plan image : position des foyers et paramètres de la conique.	42
FIG. 2.8	Translation perpendiculaire à l'axe de la caméra : le foyer d'expansion Q' et le foyer de contraction P' sont présents. .	45
FIG. 3.1	Deux intervalles de même taille ne correspondent pas à une portion d'espace de même taille.	53
FIG. 3.2	a_1 et a_2 sont chacun aussi éloigné de a_0 , mais la portion de l'espace projeté dans le segment a_1a_0 est plus petite que celle projetée en a_0a_2	53
FIG. 3.3	Le calcul est transposé du plan image sur la sphère. Les opérateurs sont exprimés sur la sphère.	54
FIG. 3.4	Sans miroir : Cône spatial vu par un support donné Ω du plan image. Avec miroir : même cône spatial vu par le point de vue «effectif» de la caméra omnidirectionnelle.	58
FIG. 3.5	Projection de l'opérateur d'un plan $Z = Z_0$ sur le plan image avec une réflexion sur le miroir.	59
FIG. 3.6	Évolution du voisinage en s'éloignant du centre.	62
FIG. 3.7	Erreur angulaire entre le vecteur flux théorique et le vecteur estimé.	67
FIG. 3.8	Direction et sens du flux optique pour une rotation R autour de l'axe de la caméra.	70
FIG. 3.9	Direction et sens du flux optique pour une translation dans l'axe de la caméra.	71
FIG. 3.10	Construction de la conique passant par un point de l'image par projection d'une droite dans la direction de translation .	73
FIG. 3.11	Direction et sens du flux optique pour une translation T dans un plan perpendiculaire l'axe de la caméra.	74

FIG. 3.12	Vecteurs flux théoriques dans le cas d'une translation perpendiculaire à l'axe de la caméra, un vecteur sur 10 est affiché.	74
FIG. 3.13	Miroir hyperboloïdal utilisé pendant les expérimentations.	76
FIG. 3.14	Dessin de gauche : opérateur symétrique. Dessin de droite : opérateur de voisinage centré à droite de l'image omnidirectionnelle.	80
FIG. 3.15	Forme en déplacement.	81
FIG. 3.16	Flux optique obtenu, un vecteur flux sur 10 est montré.	81
FIG. 3.17	Champ de vecteurs obtenus par adaptation des opérateurs avec normalisation.	83
FIG. 3.18	Champ de vecteurs théoriques normalisés.	83
FIG. 3.19	Dessin du haut : le déplacement de la forme n'affecte pas sa forme au centre de l'image. Dessin du bas : le déplacement de la forme affecte la forme à la périphérie alors qu'il s'agit de la même amplitude dans l'espace et de la même direction.	86
FIG. 3.20	Environnement créé pour la validation calcul de flux optique : <i>configuration 1</i> .	88
FIG. 3.21	Environnement créé pour la validation calcul de flux optique : <i>configuration 2</i> .	89
FIG. 3.22	<i>Configuration 1</i> : Représentation du flux optique pour une translation dans l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.	91
FIG. 3.23	<i>Configuration 1</i> : Représentation du flux optique pour une rotation autour de l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.	91
FIG. 3.24	<i>Configuration 1</i> : Représentation du flux optique pour une translation perpendiculaire à l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.	92

FIG. 3.25	<i>Configuration 2</i> : Représentation du flux optique pour une translation dans l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.	93
FIG. 3.26	<i>Configuration 2</i> : Représentation du flux optique pour une rotation autour de l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.	93
FIG. 3.27	<i>Configuration 2</i> : Représentation du flux optique pour une translation perpendiculaire à l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé. . . .	94
FIG. 3.28	Diagramme d'erreur cumulative pour une translation dans l'axe de la caméra.	97
FIG. 3.29	Diagramme d'erreur cumulative pour une rotation dans l'axe de la caméra.	98
FIG. 3.30	Diagramme d'erreur cumulative pour une translation perpendiculaire à l'axe de la caméra.	98
FIG. 3.31	Image extraite d'une séquence de translation perpendiculaire à l'axe de la caméra.	100
FIG. 3.32	Image extraite d'une séquence de translation dans l'axe de la caméra.	101
FIG. 3.33	Image extraite d'une séquence de rotation autour de l'axe de la caméra.	101
FIG. 3.34	<u>Translation dans l'axe de la caméra</u> :diagramme d'erreur cumulative.	102
FIG. 3.35	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative.	103
FIG. 3.36	<u>Translation perpendiculaire à l'axe de la caméra</u> :diagramme d'erreur cumulative.	103

FIG. 3.37	Diagramme d'erreur cumulative pour une translation perpendiculairement à l'axe de la caméra.	105
FIG. 3.38	Diagramme d'erreur cumulative pour une translation dans l'axe de la caméra.	106
FIG. 3.39	Diagramme d'erreur cumulative pour une rotation autour de l'axe de la caméra.	106
FIG. 3.40	<u>Translation perpendiculairement à l'axe de la caméra :diagramme d'erreur cumulative.</u>	107
FIG. 3.41	<u>Translation dans l'axe de la caméra :diagramme d'erreur cumulative.</u>	108
FIG. 3.42	<u>Rotation autour de l'axe de la caméra :diagramme d'erreur cumulative.</u>	108
FIG. 3.43	<u>Translation dans l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 30 et 50 pixels.</u> . .	110
FIG. 3.44	<u>Translation dans l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels.</u> .	110
FIG. 3.45	<u>Translation dans l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels</u> .	111
FIG. 3.46	<u>Translation dans l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 150 et 170 pixels</u> .	111
FIG. 3.47	<u>Translation dans l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 290 et 310 pixels</u> .	112
FIG. 3.48	<u>Translation perpendiculaire à l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels.</u>	112
FIG. 3.49	<u>Translation perpendiculaire à l'axe de la caméra :diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels</u>	113

FIG. 3.50	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 50 et 70 pixels. . .	113
FIG. 3.51	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 110 et 130 pixels .	114
FIG. 3.52	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels .	114
FIG. 3.53	<u>Translation perpendiculaire à l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels dans le cas d'images réelles.	116
FIG. 3.54	<u>Translation perpendiculaire à l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels dans le cas d'images réelles.	116
FIG. 3.55	<u>Translation perpendiculaire à l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels dans le cas d'images réelles.	117
FIG. 3.56	<u>Translation dans l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 50 et 70 pixels dans le cas d'images réelles.	117
FIG. 3.57	<u>Translation dans l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels dans le cas d'images réelles.	118
FIG. 3.58	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 30 et 50 pixels dans le cas d'images réelles.	118
FIG. 3.59	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels dans le cas d'images réelles.	119

FIG. 3.60	HSOmni : représentation du flux optique pour une translation dans l'axe de la caméra pour une image synthétique. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.	121
FIG. 3.61	<u>Translation perpendiculaire à l'axe de la caméra</u> :diagramme d'erreur cumulative.	121
FIG. 3.62	<u>Translation dans l'axe de la caméra</u> :diagramme d'erreur cumulative.	122
FIG. 3.63	<u>Rotation autour de l'axe de la caméra</u> :diagramme d'erreur cumulative.	122
FIG. 4.1	Système utilisé : deux caméras et un miroir hyperboloïdal posés sur un moteur PAN-TILT. La deuxième figure montre les deux visions imbriquées, le reflet de la caméra a été remplacé par l'image fournie par le seconde caméra.	126
FIG. 4.2	Figure illustrant les différentes rotations et les translations de la sphère, ainsi que le point projeté sur la sphère.	130
FIG. 4.3	Projection en 2D des rotations PAN et TILT du système, A_i étant la position du centre de la rotation et R le centre du repère lié au miroir.	133
FIG. 4.4	Système visuel.	134
FIG. 4.5	Détection pixel par pixel : l'opérateur bouge le bras et des pixels en mouvement indépendant sont détectés.	139
FIG. 4.6	Détection pixel par pixel : pixels en mouvement détectées et fausses détections sont présentes.	139
FIG. 4.7	Vue du dessus de la scène virtuelle.	143
FIG. 4.8	Les 100 premières images : seule la sphère est en mouvement.	144
FIG. 4.9	Entre les images 100 et 200 : le cube est en mouvement et est détecté.	144

FIG. 4.10	Entre les images 200 et 300 : la sphère et le cube sont en mouvement et sont détectés. Une fausse détection est aussi présente.	145
FIG. 4.11	Entre les images 350 et 400 : les trois objets sont en mouvement.	146
FIG. 4.12	Détection d'une personne en mouvement.	148
FIG. 4.13	Détection d'un piéton entre 1 et 2 mètres.	149
FIG. 4.14	Détection d'un piéton entre 2 et 3 mètres.	150
FIG. 4.15	Détection d'un piéton entre 3 et 4 mètres.	150
FIG. 4.16	Détection d'un piéton entre 4 et 5 mètres.	151
FIG. 4.17	Détection d'un piéton en plusieurs parties avec fausse détection dans l'arrière-plan.	151
FIG. 4.18	Détection d'un piéton avec fausse détection dans l'arrière-plan. Flux optique correspondant, seuls les pixels où l'amplitude du flux est très importante sont coloriés en noir.	152
FIG. 4.19	Détection d'un robot en déplacement entre 1 et 2 mètres à 20 cm/s.	153
FIG. 4.20	Détection d'un robot en déplacement entre 2 et 3 mètres à 50 cm/s.	154
FIG. 4.21	Détection d'un robot en déplacement entre 4 et 5 mètres à 40 cm/s.	154
FIG. 4.22	Détection pour le mouvement TILT : deux personnes bougent devant la caméra et sont détectées.	155
FIG. 4.23	Avec seulement une direction de l'espace, l'objet peut se trouver n'importe où sur la direction.	157
FIG. 4.24	L'axe de rotation PAN passe par le centre de la sphère. Le barycentre de la zone détectée fournit la bonne direction et permet le calcul de l'angle de rotation nécessaire.	158

FIG. 4.25	Détection d'une zone en mouvement (bras de l'opérateur) et photo correspondante pour une étude approfondie.	159
FIG. I.1	Calcul de la résolution d'une caméra omnidirectionnelle. . .	171
FIG. I.2	Résolution de la caméra omnidirectionnelle avec notre miroir hyperboloïdal.	172
FIG. III.1	Réglages du format et du nombre d'images par seconde . . .	175
FIG. III.2	Réglages des images.	176
FIG. III.3	Combinaison de deux images.	177
FIG. III.4	Calibration de la caméra omnidirectionnelle.	178
FIG. III.5	Les différents modes de détection.	178

LISTE DES NOTATIONS ET DES SYMBOLES

a :	Grand axe d'une conique.
b :	Petit axe d'une conique.
c :	Moitié de la distance entre les deux foyers de l'hyperbole.
(x, y) :	Coordonnées d'un point dans le plan image.
(X, Y, Z) :	Coordonnées d'un point dans l'espace.
l :	Distance du centre de la sphère au point de projection centrale dans la théorie unificatrice des caméras omnidirectionnelles.
m :	Distance du centre de la sphère au plan image dans la théorie unificatrice des caméras omnidirectionnelles.
W :	Fonction de poids dans le calcul du flux optique définissant l'importance des pixels dans un voisinage.
A :	Vitesse de rotation du miroir autour de l'axe X .
B :	Vitesse de rotation du miroir autour de l'axe Y .
C :	Vitesse de rotation du miroir autour de l'axe Z .

LISTE DES TABLEAUX

TAB. 3.1	Caractéristique de la caméra omnidirectionnelle.	76
TAB. 3.2	Amplitude et erreur en orientation du champ de vecteurs de flux avec l'algorithme adapté et le programme de OPENCV.	82
TAB. 3.3	Amplitude et erreur en orientation du champ de vecteurs de flux avec l'algorithme adapté et le programme de OPENCV. ErAmp = Erreur moyenne en amplitude, EtAmp = Écart type de l'erreur en Amplitude, ErO = Erreur en Orientation, EtO = Écart type de l'erreur en orientation.	85
TAB. 3.4	Etudes des champs de flux optique obtenus pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV, pour une translation perpendiculaire à l'axe (de façon horizontale sur l'image considérée).	95
TAB. 3.5	Etudes des champs de flux optique obtenus pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV, pour une rotation autour de l'axe de la caméra.	95
TAB. 3.6	Etudes des champs de flux optique obtenus pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV, pour une translation dans l'axe de la caméra.	95
TAB. 3.7	<u>Translation dans l'axe de la caméra</u> :études des champs de flux optique obtenues pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV.	99
TAB. 3.8	<u>Translation perpendiculaire à l'axe</u> :études des champs de flux optique obtenues pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV (translation horizontale sur l'image considérée).	99

TAB. 3.9	<u>Rotation autour de l'axe</u> :études des champs de flux optique obtenues pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV.	100
TAB. 4.1	Détection de piéton en fonction de la vitesse du piéton et la distance du piéton à la caméra.	148

LISTE DES ANNEXES

ANNEXE I	RÉSOLUTION DES SYSTÈMES CATADIOPTRIQUES .	170
ANNEXE II	JACOBIAN DE LA PROJECTION INVERSE	173
ANNEXE III	MANUEL D'UTILISATION DU PROGRAMME	174
III.1	Menu Fichier	174
III.1.1	Lancer les vidéos	174
III.1.2	Combinaison des images	175
III.1.3	Mouvement des moteurs	175
III.2	Menu Calibration	176
III.3	Menu Détection	176

INTRODUCTION

Les systèmes de vision conventionnels sont limités par leur champ de vision restreint. Est-il possible d'avoir une caméra vidéo qui puisse à chaque instant "voir" dans toutes les directions? Une telle caméra dite "omnidirectionnelle" aurait un impact sur une multitude d'applications dans le domaine de la vision par ordinateur. Une des solutions consiste à associer à une caméra directionnelle, c'est-à-dire de champ visuel limité, une surface réfléchissante pour agrandir son champ de vision. Il s'agit des systèmes catadioptriques. Cette forme de senseur apparaît pour la première fois dans la littérature dans (Rees, 1970). La sous-classe importante des caméras catadioptriques dont il est question dans ce mémoire est celle possédant un point de vue unique. Cette caractéristique assure que chaque pixel de l'image ne correspond aux rayons lumineux que d'une direction de l'espace. Dans (Nayar1, 1998), tous les systèmes catadioptriques qui satisfont la contrainte de point de vue unique sont déterminés et il y a trois surfaces réfléchissantes possibles : un miroir de forme paraboloidal combiné à une caméra munie d'une lentille orthographique, un miroir hyperboloïdal ou un miroir de forme elliptique combiné à une caméra directionnelle. Le système catadioptrique donne alors le même résultat qu'une caméra effectuant une rotation autour de son foyer, mais sans avoir à tourner la caméra. Les systèmes ainsi construits possèdent un champ de vision de 360° autour de l'axe de la caméra, comme le montre la figure 1. Cet avantage se fait au détriment d'une perte de résolution dans l'image comme nous le verrons par la suite. Comme ces systèmes possèdent un point de vue unique, nous pouvons espérer utiliser les théories déjà élaborées pour les caméras directionnelles avec ces nouveaux senseurs. Néanmoins, il est nécessaire de développer de nouveaux modèles à cause de la géométrie projective induite par la projection sur un miroir. Leur géométrie projective a été développée et utilisée depuis bientôt une décennie. Quant au champ d'application

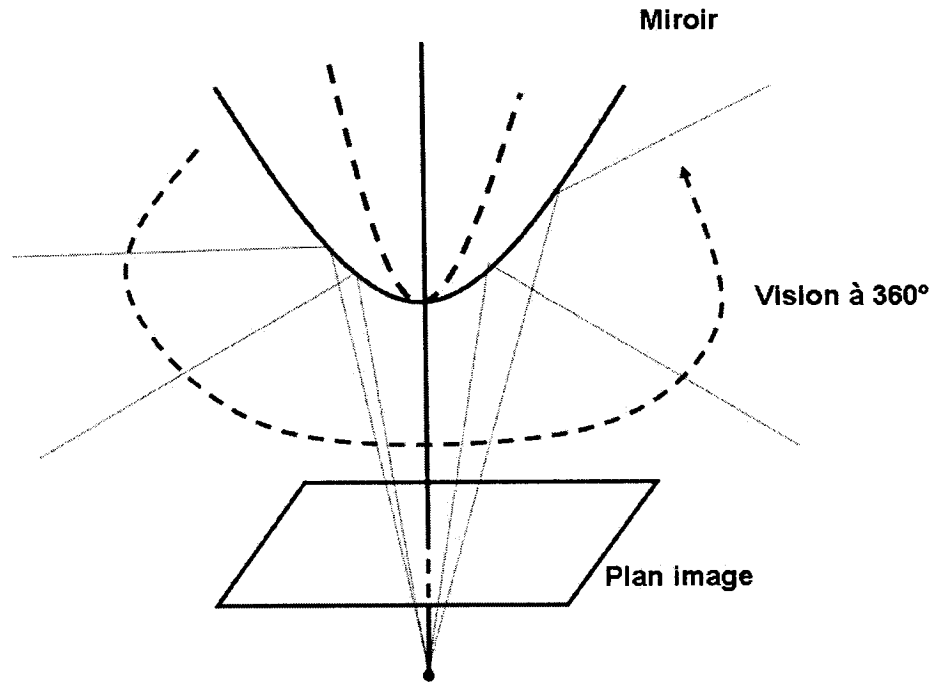


FIG. 1 Le champ de vision est de 360° autour de l'axe de la caméra.

des systèmes catadioptriques, il est vaste et comprend la navigation (Shirai, 2004), la localisation de plate-forme mobile (Yago, 2000) ou encore la surveillance vidéo (Gandhi, 2004).

L'objectif principal de notre projet est, dans un contexte dans lequel des éléments de la scène sont en mouvement et le système de capture d'information est aussi en mouvement, de pouvoir détecter, suivre et caractériser le mouvement des éléments de la scène, indépendamment du mouvement du système de capture, et ceci pour une région de l'espace aussi large que possible. L'approche adoptée est de combiner un système de vision omnidirectionnelle, constitué d'un système catadioptrique, et un système unidirectionnel, constitué d'une caméra unidirectionnelle. Les rôles

étant partagés de la manière suivante :

- le système catadioptrique détecte la présence et la direction du mouvement,
- le système directionnel est alors pointé vers l'élément en mouvements pour obtenir plus d'information grâce à sa plus grande résolution et pour un suivi de l'élément.

Comme la technique de détection choisie utilise le flux optique dans une suite d'images, il faut l'aborder dans le cas d'images omnidirectionnelles, ce qui n'a été fait que par un seul groupe de recherche dans (Daniilidis, 2002). Comme nous le verrons par la suite, (Daniilidis, 2002) a montré que les hypothèses de calcul utilisées dans le cas en perspective sont inadéquates dans le cas omnidirectionnel, et une méthode de calcul de flux optique adaptée aux images omnidirectionnelles constitue donc le deuxième objectif de notre projet de recherche.

Le premier chapitre sera consacré à l'état de l'art sur les caméras omnidirectionnelles et sur la détection de mouvement avec caméra en mouvement. Le deuxième chapitre décrit en détail la théorie des caméras omnidirectionnelles dont nous avons eu besoin pour notre étude. Le troisième chapitre sera consacré au calcul du flux optique dans une image omnidirectionnelle. Le quatrième chapitre portera sur la description du protocole de détection de mouvement développé et fournira une analyse quantitative des résultats obtenus lors de nos expérimentations.

CHAPITRE 1

POSITIONNEMENT DU SUJET

1.1 Revue Bibliographique

Nous allons tout d'abord présenter les modèles mathématiques associés aux caméras omnidirectionnelles, puis les théories inspirées du cas en perspective et adaptées au cas omnidirectionnel. Finalement, les méthodes de segmentation basée sur le mouvement appliquées aux images omnidirectionnelles seront présentées.

1.1.1 Systèmes catadioptriques

La littérature (Nayar2, 1997) définit les *systèmes catadioptriques* comme l'association d'une lentille avec une surface réfléchissante. Nous utiliserons désormais le terme de *caméra omnidirectionnelle* pour désigner les systèmes catadioptriques. Par la suite nous ferons référence aux *images en perspective* ou aux *caméras en perspective* pour évoquer les images et les caméras traditionnelles, et les différencier ainsi des images et des caméras omnidirectionnelles. Le dernier terme à définir est le *champ de vision*, qui correspond à la partie de l'espace vue par le système visuel.

1.1.1.1 Modèles mathématiques

Dans (Nayar1, 1997), S.K.Nayar pose les bases de la théorie des caméras omnidirectionnelles à miroir conique. Il prouve que sous certaines conditions, il est possible d'obtenir des systèmes visuels catadioptriques à point de vue unique. Si nous nous

plaçons dans le référentiel de la caméra, cette caractéristique assure que chaque pixel de l'image ne correspond aux rayons lumineux que d'une direction de l'espace. Comme les rayons lumineux se reflètent sur un miroir avant de toucher le plan image, l'existence d'un point de vue unique avec une caméra omnidirectionnelle n'est pas évidente. Les conditions pour avoir une caméra omnidirectionnelle à point de vue unique portent sur la forme du miroir, sur le type de caméra utilisé et sur le positionnement de la caméra par rapport au miroir. Grâce à (Nayar1, 1997), on sait qu'il existe trois cas possibles :

1. un miroir ellipsoïdal dont le second foyer correspond au foyer d'une caméra en perspective.
2. un miroir hyperboloïdal dont le second foyer correspond au foyer d'une caméra en perspective.
3. un miroir paraboloidal associé à une caméra munie d'une lentille orthographique.

Les calculs menant à l'obtention de ces géométries de miroir seront développés de façon complète dans le chapitre 2. Pour démontrer l'avantage d'avoir un point de vue unique, définissons d'abord quelques termes. Dans le cas d'une caméra munie d'une lentille en perspective, le point de vue est le point de l'espace par lequel tous les rayons lumineux émanant de la scène qui sont projetés sur le plan image. Un système catadioptrique a aussi un point de vue réel, celui de la caméra utilisée, par contre, il a aussi un autre point de vue, dit effectif vers lequel convergent tous les rayons avant de se réfléchir sur le miroir. Dans le cas d'un miroir hyperboloïdal par exemple, le point de vue effectif est le foyer de la branche de l'hyperboloïde existante. L'avantage de concevoir un système catadioptrique ayant un point de vue unique est d'obtenir une image en perspective qu'aurait obtenu une caméra en perspective si elle avait été placée au point de vue effectif. Cette caméra est dite virtuelle. La figure 1.1 illustre les différents points de vue, celui de la caméra et celui

dit effectif, dans le cas d'un miroir hyperboloïdal, ainsi que la caméra virtuelle.

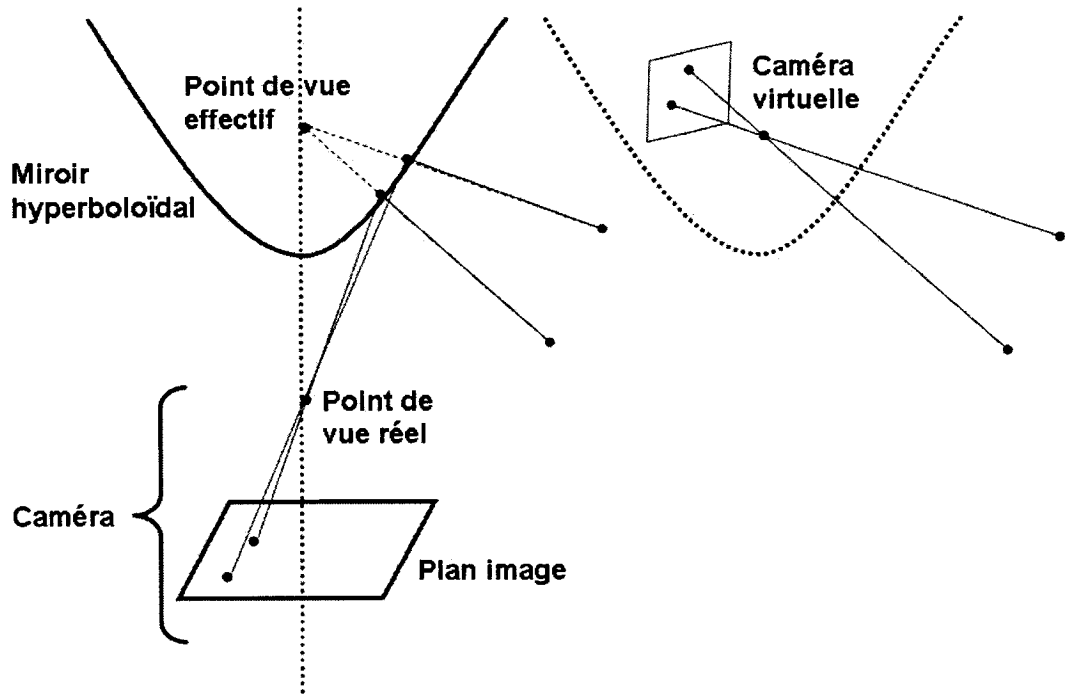


FIG. 1.1 Dans le cas d'un miroir hyperboloïdal : les rayons se dirigent vers le point de vue effectif, ici le foyer de l'hyperboloïde, se réfléchissent sur le miroir et se dirigent vers le point de vue réel de la caméra, situé au second foyer de l'hyperboloïde, et touchent le plan image.

Les équations de projection de l'espace sur le plan image d'une caméra omnidirectionnelle sont similaires pour les trois types cités précédemment et ont été développées par deux groupes de recherche :

- Le cas du miroir paraboloidal a d'abord été explicité dans (Nayar1, 1997) par S.K.Nayar et al.
- Tomas Pajdla et al. développent les équations pour tous les types de miroir dans (Pajdla, 1998).

K.Daniilidis propose une approche plus globale des caméras omnidirectionnelles. Il développe une théorie unificatrice dans (Daniilidis, 2000) qui réunit toutes les équations de projection des caméras omnidirectionnelles dans le même système d'équations. Elle est utilisée maintenant par la majorité lue dans la littérature. Elle permet d'utiliser la même équation pour toutes les formes de miroirs utilisées, en changeant seulement la valeur de certains paramètres en fonction du miroir. K.Daniilidis a démontré que pour les trois types de caméras (donc trois types de miroirs), la projection pouvait être remplacée par deux projections successives, une projection sur une sphère unitaire suivie d'une projection centrale. Cela permet d'aborder les problèmes reliés aux caméras omnidirectionnelles dans le cadre général et non pas pour un type de miroir précis. La démonstration complète comprenant la position de la sphère, du point et du plan utilisant la projection centrale sera fournie au chapitre 2. Les chercheurs ont pu dès lors développer des outils mathématiques et logiciels dans le cadre général des caméras omnidirectionnelles, et non pas pour un type de miroir précis.

1.1.1.2 Calibration des caméras omnidirectionnelles

Plaçons nous dans le cas d'une caméra en perspective avec comme centre du référentiel le centre de projection de la caméra et comme axe des Z l'axe optique de la caméra. La projection d'un point de l'espace (X, Y, Z) vers un point de l'image (x, y) peut s'écrire ainsi :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{pmatrix} \quad (1.1)$$

Si nous souhaitons maintenant obtenir ses coordonnées en pixel dans l'image $(u, v, 1)$, il faut avoir recours aux paramètres internes de la caméra :

- les coordonnées du centre de projection (u_0, v_0, w_0) ,

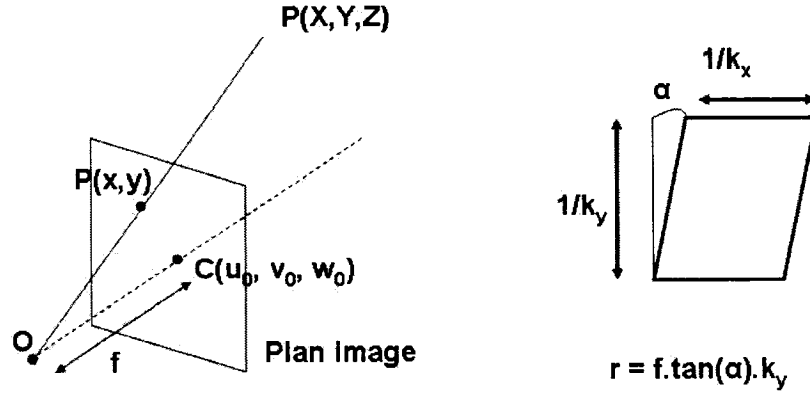


FIG. 1.2 Illustration des paramètres de la matrice de calibration.

- la distance focale f (ces systèmes respectent la contrainte du point de vue unique, et ont donc un centre de projection),
- k_x et k_y les facteurs d'échelle horizontal et vertical.
- r est le facteur de distorsion de l'image, si l'image est un rectangle, $r = 0$.

Nous retrouvons ces paramètres dans la matrice dite de calibrage K et sur la figure 1.2 :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} X/Z \\ Y/Z \\ 1 \end{pmatrix} = \begin{pmatrix} f k_x & r & u_0 \\ 0 & f k_y & v_0 \\ 0 & 0 & w_0 \end{pmatrix} \begin{pmatrix} X/Z \\ Y/Z \\ 1 \end{pmatrix} \quad (1.2)$$

Pour relier un point dans l'espace à son image dans le plan image en utilisant les coordonnées pixels, il est donc nécessaire de calculer la matrice K , c'est-à-dire les paramètres internes de la caméra.

Dans le cas d'une caméra omnidirectionnelle, la projection sur le plan image suit une équation différente. Dans le cas du miroir hyperboloïdal, l'équation de projection

sur le plan image est la suivante :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{(c^2 - a^2)}{(c^2 + a^2)Z + c\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix} \quad (1.3)$$

avec a grand axe de l'hyperboloïde et c moitié de la distance entre les deux foyers, donc les coordonnées en pixel $(u, v, 1)$ sont :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} fk_x & r & u_0 \\ 0 & fk_y & v_0 \\ 0 & 0 & w_0 \end{pmatrix} \begin{pmatrix} \frac{(c^2 - a^2)}{(c^2 + a^2)Z + c\sqrt{X^2 + Y^2 + Z^2}} X \\ \frac{(c^2 - a^2)}{(c^2 + a^2)Z + c\sqrt{X^2 + Y^2 + Z^2}} Y \\ 1 \end{pmatrix} \quad (1.4)$$

Les différentes méthodes dites de calibration se classent selon trois groupes distincts.

- En connaissant des points dans l'espace : ce genre de calibration utilise des points dans l'espace connus et dont la projection sur le point image est facilement détectable (coin ou patrons de calibration). (Aliaga, 2001) propose une calibration en deux étapes : le centre de projection est estimé manuellement, puis un ensemble de points dans l'espace dont la distance à la caméra est connue sont utilisés pour obtenir le reste des paramètres. (Fabrizio, 2002) met en place une calibration utilisant deux cercles dans l'image comme ensemble de points connus. Une aiguille en forme de cône est placée au sommet du miroir et ses contours créent sur l'image un premier cercle. Le cercle extérieur de l'image omnidirectionnelle produit par la limite extérieure du miroir fournit le deuxième cercle. Ces deux cercles sont issus de deux plans dont les positions spatiales sont connues. (Shirai, 2004) commence par estimer le centre de projection et les facteurs d'échelle en utilisant des points sur ce cercle. Un patron de calibration dont la position spatiale est connue est utilisée pour calculer la distance focale.
- Auto-Calibration : ce genre de calibration utilise seulement la correspondance de points dans plusieurs images sans connaître la position de points dans l'es-

pace ou la position de la caméra. Dans (Kang, 2000) et dans (Pajdla, 2004), des méthodes de calibration s'appuient sur la géométrie épipolaire et permettent une calibration fiable dans le cas d'un miroir paraboloidal. Le caractère automatique de (Kang, 2000) repose sur un algorithme pour détecter des points faciles à identifier d'une image à l'autre. La calibration utilise des paires d'images qui présentent des points correspondant au même objet. Pour chaque paire de points, les courbes épipolaires sont déterminées dans les images et l'erreur se calcule à partir de la distance du point à la courbe. (Pajdla, 2004) utilise un algorithme RANSAC(Random Sample Consensus) à neuf points (Fischler, 1981) pour identifier des paires de points.

- Projections de droites : ce genre de méthode utilise les images de droites de l'espace 3D dans le plan image sans aucune donnée métrique. Dans (Daniilidis, 2002), la méthode utilise le fait que la projection d'une droite de l'espace produit un arc de cercle dans l'image dans le cas d'un miroir paraboloidal (les détails seront fournis au chapitre 2). Si cet arc est une ellipse, la transformation qui la fait passer de l'ellipse au cercle va permettre de trouver les facteurs d'échelle vertical et horizontal. La position du centre du miroir et de la distance focale sont trouvés à partir de propriétés géométriques des images des droites. (Barreto, 2003) propose une méthode plus précise de calibration à partir d'images de droites de l'espace en deux étapes : le centre de projection est estimé en utilisant les intersections de trois projections de droites. Ensuite, le centre de projection ainsi calculé est utilisé pour déterminer l'image d'une conique dite absolue (projection d'un plan situé à l'infini) et permet d'obtenir les autres paramètres intrinsèques. Pour accomplir ces calculs, une bonne estimation des coniques, est requise. Il propose donc une méthode d'estimation plus précise que celle de (Daniilidis, 2002), et fournit ainsi une meilleure calibration.

Dans (Ying, 2004), deux équations de contrainte sont exprimées pour l'image d'une droite dans le plan image et trois pour l'image de sphère dans le plan

image. Il utilise les projections de droites dans le plan image mais calcule aussi les projections de sphères sur le plan image. Il développe ainsi une procédure de calibration qui nécessite deux droites ou trois sphères. Contrairement aux articles précédents, cet article se place dans le cadre général des caméras omnidirectionnelles grâce à la théorie unificatrice de (Daniilidis, 2000), dans laquelle l'image d'un segment de droite est un segment de conique (dans le cas paraboloidal, c'est un cercle). Ce segment de conique est difficile à paramétrer. L'image d'une sphère est aussi une conique, mais cette fois-ci complète, et est donc plus facile à estimer. La procédure de calibration avec des sphères donne des résultats plus précis.

1.1.1.3 Calcul du flux optique dans les images omnidirectionnelles

Le calcul du flux optique dans des images omnidirectionnelles a été étudié par deux groupes seulement. Tout d'abord dans (Stratmann, 2002), cinq techniques de calcul de flux optique sont testées et comparées sur des images omnidirectionnelles. Il en ressort que la technique différentielle avec hypothèse de constance locale attribuée à Lucas et Kanade (Lucas, 1981) présente le meilleur compromis entre vitesse de calcul et taux d'erreur. Dans (Daniilidis, 2002), l'utilisation «aveugle» des opérateurs habituellement utilisés sur des images en perspective est critiquée. Quand il faut faire des calculs en un pixel de l'image, certains calculs, comme des dérivées spatiales de l'image, s'appuient sur la valeur des pixels dans un voisinage autour de ce pixel. Les opérateurs usuels utilisent des voisinages carrés centrés sur ce pixel. Dans l'annexe I, il est démontré que la résolution évolue en fonction de la position dans l'image. Ils ne reflètent pas les déformations locales présentes dans une image omnidirectionnelle qui implique cette résolution variable.

Pour (Daniilidis, 2002), les calculs doivent se faire dans un ensemble dans lequel

les anciennes hypothèses sont encore valides, comme par exemple la surface des miroirs. Pour donner un cadre plus général à son calcul, il utilisera la théorie unificatrice développée dans (Daniilidis, 2000) et utilisera des projections sphériques qui seront décrites dans le chapitre 2. Il adapte une technique différentielle dite de constance locale, aussi appelée technique de Lucas et Kanade (Lucas, 1981). Pour tester l'algorithme, une caméra omnidirectionnelle utilisant un miroir paraboloidal subit une translation pure et l'erreur est estimée sur la direction de translation calculée à partir du flux optique pour différentes techniques de calcul de flux optique. La technique qui utilise la sphère virtuelle permet une légère amélioration des résultats par rapport à la technique de Lucas et Kanade.

1.1.1.4 Calcul du mouvement propre avec une caméra omnidirectionnelle

L'avantage des caméras omnidirectionnelles est que leur large champ de vision permet de voir des éléments de mouvement qui ne se manifestent pas toujours dans des petits champs de vision. (Stratmann, 2002) montre que dans le cas de translations perpendiculaires à l'axe de la caméra (illustration sur la figure 1.3), il est possible de détecter dans une même image le foyer d'expansion (point duquel tous les points de l'image semblent s'éloigner) et le foyer de contraction (point vers lequel tous les points de l'image semblent se diriger). Des groupes de recherche ont voulu exploiter cette perception différente du mouvement pour une caméra omnidirectionnelle en mouvement en calculant le mouvement propre de la caméra. Le mouvement propre est la transformation spatiale subie par la caméra et possède deux composantes, une translation T et une rotation R . La plupart des méthodes comportent deux étapes :

1. le flux optique est calculé pour deux images,

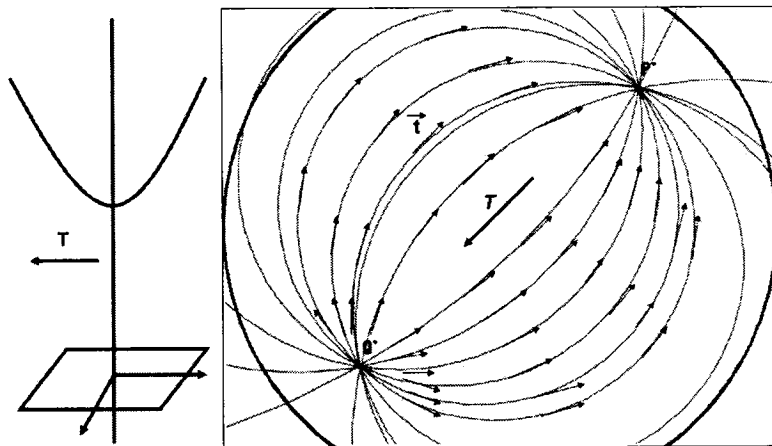


FIG. 1.3 Translation perpendiculaire à l'axe de la caméra : le foyer d'expansion Q' et le foyer de contraction P' sont présents.

2. R et T sont extraits du flux pour une série de points reliés d'une image à l'autre.

En plaçant une caméra sur une plate forme mobile, le calcul de mouvement propre permet de connaître le déplacement de la plate forme à partir des seules images.

Le calcul du mouvement propre pour des images omnidirectionnelles a été considéré par quatre groupes.

- Dans le cadre d'un miroir paraboloidal, le calcul est transposé sur une sphère virtuelle dans (Nayar2, 1998). Une fois le flux optique calculé, deux techniques existantes pour les images en perspective sont adaptées au cas de la sphère : l'algorithme de Bruss-Horn (Horn, 1983) et l'algorithme Heeger-Jepson (Jepson, 1992).
- Dans le cas spécifique d'un miroir hyperboloïdal, (Gandhi, 2003) utilise une adaptation de la méthode d'estimation du mouvement propre dite directe. Elle uti-

lise l'équation de contrainte du flux optique et les équations de projection de la caméra omnidirectionnelle pour estimer le mouvement propre à partir d'un nombre fini de points sur le sol. Comme la contrainte du flux optique n'est satisfaite que pour des faibles déplacements dans l'image, une approche multi-résolution est utilisée. Les paramètres du mouvement propre sont d'abord calculés au niveau de résolution le plus faible et une pseudo-image est ainsi générée en prenant l'image initiale et lui imposant le mouvement propre calculé. Le mouvement propre est calculé au niveau supérieur à partir de l'image initiale et de la pseudo-image, et la procédure est ainsi réitérée jusqu'au niveau de résolution le plus élevé. Le processus est robuste et précis, mais ne fonctionne pas encore en temps réel.

- Le calcul du mouvement propre va être reconsidéré avec l'apparition de la théorie unificatrice (Daniilidis, 2000), puisque le calcul sur une sphère rentre directement dans le cadre de cette théorie qui fait de la sphère une des étapes de la projection sur le plan image. Dans (Vassallo, 2002) et (Sastrey, 2003), un calcul de mouvement propre est proposé quel que soit le type de miroir utilisé, en se servant des deux algorithmes précédemment cités ((Horn, 1983) et (Jepson, 1992)).

1.1.2 Segmentation basée sur le mouvement dans le cadre d'une caméra omnidirectionnelle

Nous allons d'abord définir certains termes reliés à l'étude des systèmes mobiles qui seront utilisés dans le reste de ce mémoire. Le terme de *segmentation de mouvement dans une séquence d'images* désigne l'action de représenter une séquence d'images en terme de régions animées par un mouvement cohérent. La détection de mouvement est un cas spécial de la segmentation de mouvement qui sépare les régions en deux groupes :

- régions en mouvement contre régions statiques dans le cas d'une caméra statique,

- mouvement propre contre mouvement indépendant dans le cas d’une caméra mobile.

Ce mémoire se place dans le second cas et nous utiliserons le terme *arrière-plan* pour définir l’environnement statique, et le terme *objets en mouvement indépendants* pour désigner le reste de la scène. Le terme de *segmentation basée sur le mouvement* désigne les techniques de segmentation qui utilisent le mouvement apparent dans l’image, c’est-à-dire le flux optique, comme première étape de leur segmentation. Si la segmentation basée sur le mouvement est un domaine de recherche qui n’est pas récent, son application aux caméras omnidirectionnelles n’a été abordée que par quelques groupes de recherche.

La première approche est développée dans (Stratmann, 2003). La segmentation s’effectue dans le cas d’une plate-forme mobile avec une caméra omnidirectionnelle posée à son sommet, se déplaçant sur un sol supposé plat. Les mouvements de la plate-forme sont donc des rotations selon un axe et des translations dans un plan unique. Alors que l’arrière-plan représente la plus grande partie du champ de vecteurs flux optique dans l’image, des objets en mouvement indépendants apparaissent comme des régions de perturbations dans l’image. La méthode suivante est appliquée.

- Le flux optique est calculé à partir des deux premières images.
- Une pseudo-image est générée à partir de la seconde image en utilisant le flux optique calculé.
- La pseudo image est comparée à la troisième image et les régions de différence maximale sont détectées.

Ces régions sont soit des objets en mouvement indépendants, soit des objets très proches de la caméra qui font varier la valeur du flux en amplitude mais pas en orientation.

La deuxième approche a été effectuée dans (Gandhi, 2003). Le mouvement propre de la caméra posée sur une voiture électrique est tout d’abord calculé. Le sol est approximé par une surface plane et le mouvement propre est calculé sur cette surface. Après compensation du mouvement propre, les caractéristiques sur le sol seront alignées dans deux images successives alors que celles dues aux objets stationnaires ou en mouvement sur le sol ne le seront pas. La différence entre les deux images mettra en évidence les objets et supprimera les caractéristiques du sol. Comme dans (Stratmann, 2003), la différence entre les deux images dépend autant du mouvement résiduel dans le cas des objets en mouvement, que des gradients spatiaux dans le cas des objets statiques. Pour ne garder finalement que les objets en mouvement, la différence entre les images est *normalisée* selon la technique explicitée dans (Trucco, 1998). Un suivi est alors effectué pour chaque zone détectée. Pour une caméra posée sur une voiture électrique, le système donne de bons résultats et détecte piétons et véhicules passant à côté de la caméra. La même technique est utilisée dans (Gandhi, 2004), mais avec deux autres plate-formes. Tout d’abord, un robot mobile se déplaçant lentement fournit de bonnes détections. Ensuite une caméra omnidirectionnelle montée sur un casque, et posée sur la tête d’un individu, fournit des bonnes détections mais le suivi des zones en mouvement est perturbée par les rotations de large amplitude du système ainsi construit.

Le troisième type de segmentation réalisé avec des images omnidirectionnelles dans le cadre d’une coopération de plusieurs robots mobiles est décrit dans (Sastry, 2003). Le but final est une stratégie de contrôle de formation d’une flotte de robots mobiles. L’idée est de pouvoir détecter tous les robots en mouvement pour tout d’abord éviter les collisions et ensuite reconnaître le meneur et le suivre dans ses déplacements. Chaque robot est doté d’une caméra omnidirectionnelle. C’est là aussi un cas de déplacement sur un sol supposé plat. Le mouvement apparent dans l’image, le flux optique, est relié au mouvement dans la scène 3D dans ce

cas de mouvements rigides simples. Les résultats sont stockés pour les n pixels de l'image pour m images dans une matrice dite de flux optique W . Une analyse en composantes principales est appliquée sur la matrice W et permet de déterminer $k + 1$ ensembles de pixels, pour les k robots dans l'image et l'arrière plan. Chaque pixel est ainsi rattaché à l'ensemble qui lui est le plus proche. Cette technique suppose de connaître le nombre de robots dans l'image.

1.2 Description du projet

Pour reprendre les objectifs énumérés en introduction de ce mémoire, la finalité de ce projet était de construire un système visuel alliant les avantages d'une vision omnidirectionnelle et ceux d'une vision directionnelle. Dans le cas où des éléments de l'espace se déplacent et que le système de capture d'images est lui aussi en mouvement, il s'agit de détecter ces éléments en mouvement et de les caractériser. Ce système visuel serait en charge une détection de mouvement dans une région de l'espace aussi grande que possible, d'où le recours à une caméra omnidirectionnelle constituée d'un système catadioptrique, mais serait aussi en charge une récolte d'information sur la zone en mouvement grâce à une caméra directionnelle. Nous pouvons dire que chaque type de vision compense les lacunes de l'autre. La caméra omnidirectionnelle perçoit plus de mouvement grâce à son champ de vision plus large que celui de la directionnelle. La partie directionnelle apporte une grande résolution qui compense la faible résolution de l'omnidirectionnelle dans certaines parties de l'image.

Nous allons développer la problématique de ce projet puis nous énumérerons un par un les objectifs de ce projet.

1.2.1 Problématiques

La problématique principale repose sur la mise en place pour le système d'une détection de mouvement efficace. Il s'agit de différencier dans l'image le mouvement propre de la caméra et le mouvement dû aux objets se déplaçant indépendamment. Pour effectuer une segmentation qui s'appuie sur le flux optique, comme décrite dans la section 1.1.2, la difficulté consiste à d'abord exprimer les modèles de mouvement dans le cadre des caméras omnidirectionnelles. Dans le cas en perspective, avec comme centre du référentiel le point de vue de la caméra et comme axe de Z, l'axe optique de la caméra, pour un point $P = (X, Y, Z)$ de l'espace, l'équation de projection est :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{pmatrix} \quad (1.5)$$

Le mouvement 2D dans l'image induit par le mouvement 3D rigide d'un objet dans l'espace, s'élabore comme suit :

$$\dot{P} = -T - R \times P \quad (1.6)$$

avec \dot{P} la dérivée temporelle de P , $R = (R_x, R_y, R_z)$ la vitesse angulaire et $T = (T_x, T_y, T_z)$ la vitesse de translation. À partir de là, les équations 1.5 et 1.6 fournissent le modèle de mouvement :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -R_x xy + R_y(x^2 + 1) - R_z y + \frac{T_x - T_z x}{Z} \\ -R_x(y^2 + 1) + R_y xy - R_z x + \frac{T_y - T_z y}{Z} \end{pmatrix} \quad (1.7)$$

Dans le cas des images omnidirectionnelles, l'équation 1.5 devient (cas d'un miroir

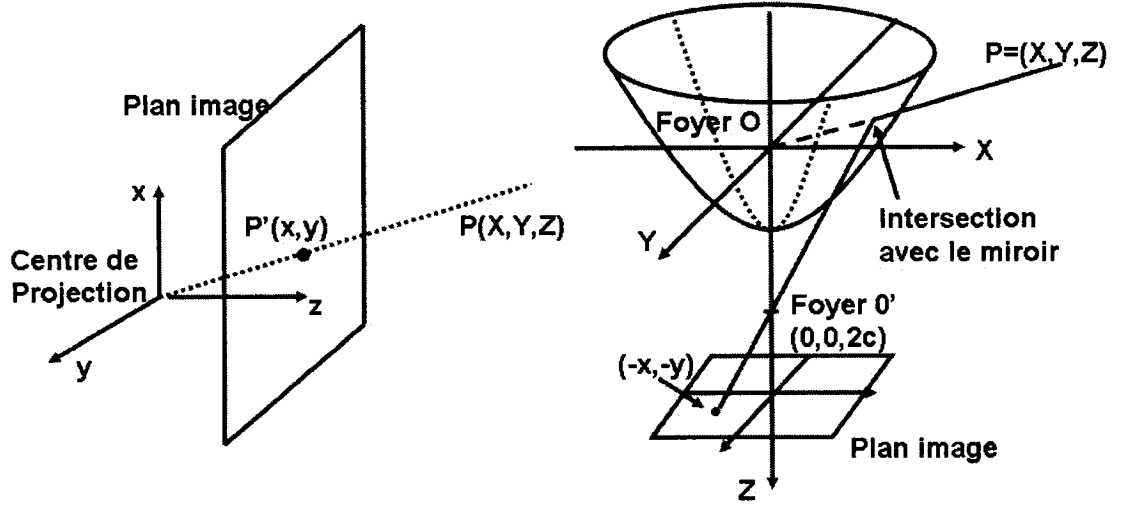


FIG. 1.4 Deux types de projection sur le plan image.

hyperboloïdal) :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{(c^2 - a^2)}{(c^2 + a^2)Z + c\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix} \quad (1.8)$$

avec a grand axe de l'hyperboloïde et c moitié de la distance entre les deux foyers. Les deux projections sont illustrées sur la figure 1.4. Pour les caméras omnidirectionnelles, l'équation équivalente à l'équation 1.7 n'est pas développée dans le cas d'un mouvement quelconque de la caméra. Les systèmes catadioptriques ont été utilisés principalement au sommet de plate formes mobiles avec comme axe optique une droite perpendiculaire au sol. Les seuls mouvements considérés sont alors des mouvements de rotation autour de l'axe, et des mouvements de translation perpendiculaire à l'axe de la caméra comme illustré sur la figure 1.5. Même dans ces deux cas très simples, il n'est pas possible d'écrire un modèle de mouvement (c'est-à-dire relier dans une même équation (\dot{x}, \dot{y}) et (x, y) sans faire apparaître X, Y ou Z)) Dans le cadre omnidirectionnel, l'utilisation de la théorie unificatrice de (Daniilidis, 2000) a permis de contourner le problème pour le calcul du mouve-

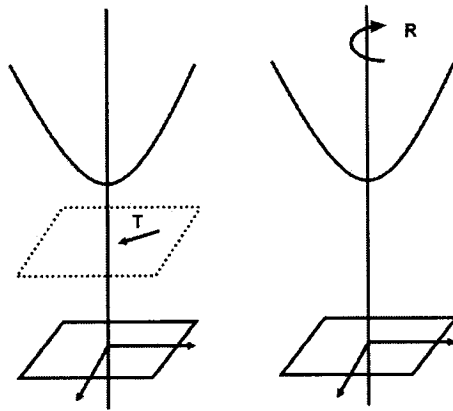


FIG. 1.5 Types de mouvement considérés dans la littérature.

ment propre de plate-formes mobiles ((Vassallo, 2002), (Sastry, 2003) et (Daniilidis, 2005)). Il utilise des projections sphériques qui seront décrites au chapitre 2 et écrit des modèles de mouvement sur la sphère au lieu du plan image.

Nous avons choisi d'effectuer une segmentation basée sur le mouvement, donc qui utilise le flux optique dans une séquence d'images. Comme vu dans la section 1.1.1.3, certains chercheurs ont mis en avant les problèmes liés aux techniques utilisées pour des images en perspective dans leurs applications au cas omnidirectionnel, mais peu de solutions sont présentées.

Pour finir, la détection de mouvement et un meilleur calcul de flux optique nécessite une bonne maîtrise de la géométrie projective des caméras omnidirectionnelles. Néanmoins, certaines des études théoriques dont nous comptons nous servir ne sont pas développées pour les miroirs hyperboloïdaux, mais seulement pour des miroirs paraboloidaux. Il s'agit alors d'effectuer les développements pour l'hyperboloïde mais aussi dans le cadre de la théorie unificatrice, c'est-à-dire de s'assurer que nos développements pourraient servir pour n'importe quel type de miroir.

1.2.2 Objectifs

L'objectif principal de notre projet est, dans un contexte dans lequel des éléments de la scène sont en mouvement et le système de capture d'information est aussi en mouvement, de pouvoir détecter, suivre et caractériser le mouvement des éléments de la scène, indépendamment du mouvement de système de capture, quelque soit la direction de l'environnement où se situent ces éléments en mouvements. L'approche adoptée est de combiner un système de vision omnidirectionnelle, constitué d'un système catadioptrique, et un système unidirectionnel, constitué d'une caméra unidirectionnelle. Les rôles étant partagés de la manière suivante :

- le système catadioptrique détecte la présence et la direction du mouvement grâce à une méthode de flux basée sur le flux optique,
- le système directionnel est alors pointé vers l'élément en mouvement pour obtenir une image en perspective à plus grande résolution et pour un suivi éventuel de l'élément.

Notre travail s'est donc orienté autour de trois axes, le développement théorique sur les caméras omnidirectionnelles utilisant un miroir hyperboloïdal, un calcul de flux optique adapté aux caméras omnidirectionnelles et la mise en place d'un système mobile de détection de mouvement. Chacun de ses axes est composé de plusieurs sous-objectifs.

Développements théoriques sur les systèmes catadioptriques :

- Une étude plus approfondie des projections de droites de l'espace vers le plan omnidirectionnel.
- L'étude des mouvements des points de l'image correspondant à des éléments immobiles de la scène en fonction des mouvements de la caméra omnidirectionnelle. Il s'agit du calcul d'un modèle de mouvement pour un système catadioptrique en mouvement.

Calcul du flux optique :

- L'étude de l'orientation théorique du flux optique dans une séquence d'images lorsque le système catadioptrique est en mouvement.
- Une méthode de calcul de flux optique mieux adaptée aux images omnidirectionnelles qui passera par une adaptation des opérateurs utilisés.

Système de détection de mouvement :

- La mise en place d'un protocole de détection de mouvements avec un système catadioptrique en mouvement.
- L'obtention d'information sur l'élément en mouvement à partir de l'image omnidirectionnelle comme sa position dans l'espace ou sa vitesse de déplacement.
- Pointer la caméra unidirectionnelle vers l'élément en mouvement.

Nous finirons ce chapitre par certaines considérations techniques nécessaires à la mise en place matérielle et logicielle de notre système.

1.2.3 Considérations techniques

1.2.3.1 Partie montage

Pour réaliser notre système de vision, nous avons employé deux caméras mises dos à dos, l'une en direction du miroir hyperboloïdal, l'autre possédant le même axe optique mais pointant dans le sens opposé.

Dans le but de tester notre segmentation, nous avons aussi eut besoin de créer un système mobile. Un double moteur donnant des mouvements de tangage et de lacet a été utilisé. Il permettra de faire évoluer le système suivant deux rotations comme illustré sur la figure 1.6. La dernière considération releva de la calibration du système omnidirectionnel, car comme nous le verrons la position de la caméra par rapport au miroir est sujette à des contraintes géométriques dans le cas du

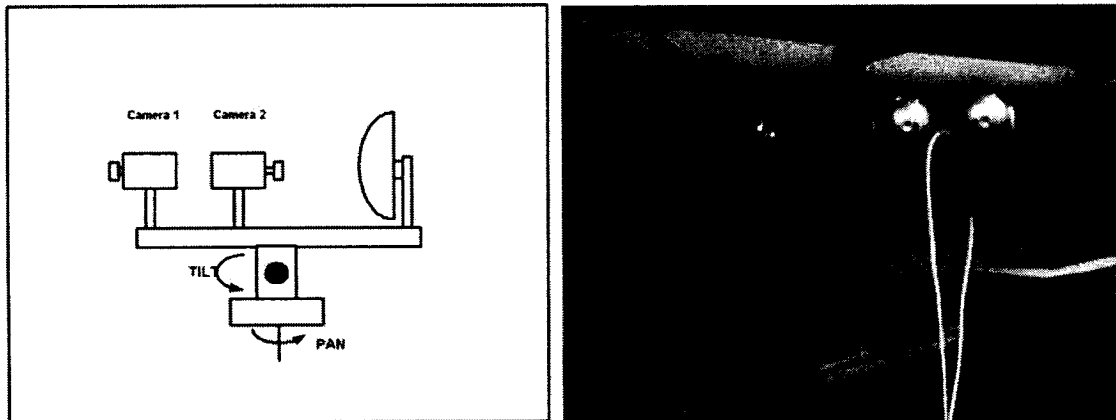


FIG. 1.6 Système utilisé : deux caméras et un miroir hyperboloïdal posés sur un double moteur

miroir hyperboloïdal.

1.2.3.2 Partie logicielle

La partie logicielle peut se décomposer sous la forme de plusieurs modules :

- Calibration des caméras utilisées : calcul des paramètres intrinsèques.
- Communication avec les moteurs : donner des mouvements calibrés en vitesse et en amplitude aux caméras.
- Traitement des images en temps réel : segmentation et identification des zones en mouvement indépendant.
- Affichage : images issues des deux caméras et affichage des zones détectées.

Maintenant que les objectifs de notre projet sont définis, le chapitre suivant va présenter les détails de la géométrie projective des caméras omnidirectionnelles.

CHAPITRE 2

BASE DE LA THÉORIE DES CAMÉRAS OMNIDIRECTIONNELLES

L'utilisation massive des caméras omnidirectionnelles est due aux travaux de S.K. Nayar (Nayar1, 1998) qui a pu prouver l'existence du point de vue unique dans le cadre des miroirs coniques et qui a ainsi pu écrire les équations de projections de ces caméras.

Nous allons dans ce chapitre décrire la géométrie projective des caméras omnidirectionnelles. Nous commencerons par justifier les différentes formes de miroirs possibles avant de décrire en détail la théorie unificatrice déjà évoquée. Nous finirons avec une section sur la projection de droites de la scène 3D sur les images omnidirectionnelles, laquelle fait partie des aspects conceptuels de notre travail.

2.1 Point de vue unique et forme des miroirs

La contrainte du point de vue unique permet d'assurer que le rayon de lumière qui traverse le capteur ne provient que d'une direction de l'espace. Il s'agit d'un système catadioptrique, c'est-à-dire avec une reprojection des rayons lumineux, il faut considérer deux points importants. Le point de vue effectif, qui correspond au point vers lequel les rayons lumineux se dirigent avant de toucher la surface réfléchissante, et le trou d'épingle effectif, qui correspond au point à travers lequel la projection en perspective sera réalisée.

2.1.1 Calcul de l'équation de contrainte avec une caméra en perspective

Commençons par considérer que le point de vue effectif v est le centre de notre système de coordonnées cartésiennes. Nommons maintenant notre trou d'épingle effectif p , et prenons l'axe des z suivant la direction \overrightarrow{vp} . Comme la projection en perspective présente une symétrie de révolution autour de tout axe passant par p , le miroir peut être considéré comme une surface de révolution autour de l'axe des z . Nous allons travailler en deux dimensions dans la base (v, \vec{r}, \vec{z}) . Nous appellerons c la distance entre v et p . Toutes ces données sont notées dans la figure 2.1.

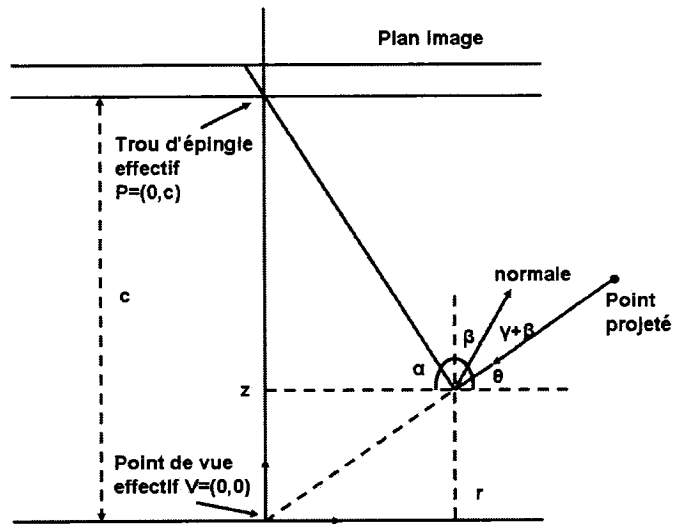


FIG. 2.1 Géométrie utilisée pour déterminer les formes de miroirs possibles dans le cadre d'un point de vue unique.

Supposons qu'un rayon touche la surface du miroir au point (r, z) formant avec l'axe r un angle θ . Si le rayon passe par le point v alors la relation suivante est

vérifiée :

$$\tan \theta = \frac{z}{r} \quad (2.1)$$

Si α est l'angle entre le rayon réfléchi sur la surface et la direction $-\vec{r}$, alors :

$$\tan \alpha = \frac{c - z}{r} \quad (2.2)$$

puisque le rayon réfléchi doit passer par le trou d'épingle $p(0, c)$. Enfin si β est l'angle entre l'axe des z et la normale au miroir au point (r, z) , alors :

$$\frac{dz}{dr} = -\tan \beta \quad (2.3)$$

Dans le cas d'un miroir parfait l'angle d'incidence est égal à l'angle réfléchi. Si γ est l'angle entre le rayon réfléchi et l'axe z , nous avons $\gamma = 90 - \alpha$ et $\theta + \alpha + 2\beta + 2\gamma = 180$. En éliminant γ de ces deux expressions, nous obtenons $2\beta = \alpha - \theta$. En prenant la tangente des deux côtés, nous obtenons finalement :

$$\frac{2 \tan \beta}{1 - \tan^2 \beta} = \frac{\tan \alpha - \tan \theta}{1 + \tan \alpha \tan \theta} \quad (2.4)$$

En se servant des équations 2.1, 2.2 et 2.3, nous obtenons finalement l'équation de contrainte du point de vue unique :

$$r(c - 2z)\left(\frac{dz}{dr}\right)^2 - 2(r^2 + cz - z^2)\frac{dz}{dr} + r(2z - c) = 0 \quad (2.5)$$

2.1.2 Solution générale de l'équation de contrainte

Il s'agit d'une équation du second degré en $\frac{dz}{dr}$ qui peut s'écrire :

$$\frac{dz}{dr} = \frac{(z^2 - r^2 - cz) \pm \sqrt{r^2 c^2 + (z^2 + r^2 - cz)^2}}{r(2z - c)} \quad (2.6)$$

Si $y = z - \frac{c}{2}$ et $b = \frac{c}{2}$, alors :

$$\frac{dy}{dr} = \frac{(y^2 - r^2 - b^2) \pm \sqrt{4r^2b^2 + (y^2 + r^2 - b^2)^2}}{2ry} \quad (2.7)$$

Ensuite, substituons $2rx = y^2 + r^2 - b^2$, et nous avons enfin :

$$\frac{1}{\sqrt{b^2 + x^2}} \frac{dx}{dr} = \pm \frac{1}{r} \quad (2.8)$$

En intégrant des deux côtés par rapport à r :

$$\ln(x + \sqrt{b^2 + x^2}) = \pm \ln r + C \quad (2.9)$$

où C est une constante d'intégration. Finalement,

$$x + \sqrt{b^2 + x^2} = \frac{k}{2} r^{\pm 1} \quad (2.10)$$

avec $k = 2e^C > 0$ une constante. Deux cas peuvent être séparés :

$$\begin{aligned} \sqrt{b^2 + x^2} &= \frac{k}{2} r - x & (k \geq 2) \\ \sqrt{b^2 + x^2} &= \frac{k}{2r} - x & (k > 0) \end{aligned}$$

En effet, dans le premier cas, nous avons besoin de $(k \geq 2)$ pour pouvoir écrire l'équation car $r \geq x$, et le deuxième cas ne présente pas de restriction (C quelconque donc $(k > 0)$). En utilisant $rx = \frac{1}{2}(y^2 + r^2 - b^2)$:

$$\begin{aligned} \frac{k}{2}(y^2 + r^2) &= \frac{k^2}{4}r^2 - \frac{c^2}{8}(2 - k) & (k \geq 2) \\ \frac{2b^2r^2}{k} &= \frac{k}{2} - y^2 - r^2 + b^2 & (k > 0) \end{aligned}$$

En introduisant maintenant $y = z - \frac{c}{2}$ et $b = \frac{c}{2}$, deux équations sont obtenues :

$$(z - \frac{c}{2})^2 - r^2(\frac{k}{2} - 1) = \frac{c^2}{4}(\frac{k-2}{k}) \quad (k \geq 2) \quad (2.11)$$

$$(z - \frac{c}{2})^2 + r^2(1 + \frac{c^2}{2k}) = \frac{2k + c^2}{4} \quad (k > 0) \quad (2.12)$$

Ces deux équations représentent l'ensemble des miroirs qui satisfont la contrainte du point de vue unique. Comme nous le verrons, si certains cas sont possibles théoriquement, ils ne sont pas réalisables en pratique.

2.1.2.1 Miroir plan

Dans l'équation 2.12, si avec $k = 2$ et $c > 0$, la solution est $z = \frac{c}{2}$. La seule possibilité est le plan médian entre v et p . Dans tous les cas, un miroir plan n'augmentera jamais le champ de vision.

2.1.2.2 Miroir en forme de cône

Dans l'équation 2.12, avec $c = 0$ et $k \geq 2$, il s'agit d'un miroir conique de section circulaire avec :

$$z = \sqrt{\frac{k-2}{2}} r^2 \quad (2.13)$$

L'angle d'ouverture du cône est 2τ avec $\tan \tau = \sqrt{\frac{2}{k-2}}$. La solution pourrait apparaître raisonnable mais comme $c = 0$, le trou d'épingle de la caméra doit être au sommet du cône, ce qui n'est pas viable.

2.1.2.3 Miroir Sphérique

Dans l'équation 2.12, en prenant $c = 0$ et $k > 0$, il s'agit d'un miroir sphérique avec :

$$z^2 + r^2 = \frac{k}{2} \quad (2.14)$$

Comme le trou d'épingle et le point de vue coïncident, cette solution est elle aussi inutilisable. Si la sphère a souvent été utilisée pour élargir le champ de vision, le trou d'épingle était placé en dehors de la sphère, et il n'y avait alors pas de point de vue unique effectif.

2.1.2.4 Miroir ellipsoïdal

Dans l'équation 2.12, quand $k > 0$ et $c > 0$, il s'agit d'un miroir ellipsoïdal :

$$\frac{1}{a_e} \left(z - \frac{c}{2} \right)^2 + \frac{1}{b_e^2} r^2 = 1 \quad (2.15)$$

avec

$$a_e = \sqrt{\frac{2k + c^2}{4}} \quad (2.16)$$

$$b_e = \sqrt{\frac{k}{2}} \quad (2.17)$$

Le trou d'épingle et le point de vue effectif sont donc placés aux foyers de l'ellipse. Le champ de vision est ainsi agrandi si la partie de l'ellipsoïde considérée est située en dessous du point de vue effectif.

2.1.2.5 Miroir hyperboloïdal

Dans l'équation 2.12, quand $k > 2$ et $c > 0$, il s'agit du miroir hyperboloïdal suivant :

$$\frac{1}{a_h}(z - \frac{c}{2})^2 - \frac{1}{b_h^2}r^2 = 1 \quad (2.18)$$

avec

$$a_h = \frac{c}{2} \sqrt{\frac{k-2}{k}} \quad (2.19)$$

$$b_h = \frac{c}{2} \sqrt{\frac{2}{k}} \quad (2.20)$$

Là aussi, nous avons une solution viable pour un système omnidirectionnel. La courbure du miroir augmente avec k et le cas limite $k \rightarrow 2$ correspond au miroir plan.

2.1.3 Calcul de l'équation de contrainte dans le cas d'une lentille orthographique

Nous avons pour l'instant seulement traité le cas des caméras en perspective, mais il existe aussi un dispositif de caméra omnidirectionnelle utilisant une lentille orthographique. La projection orthographique présentant elle aussi une symétrie axiale autour de l'axe des z , il reste juste à déterminer la fonction $z = f(r)$ comme dans le cas précédent. Chaque rayon de lumière se dirige vers le point de vue unique v et nommons θ l'angle entre le rayon incident et la normale à la surface du miroir. Nous avons donc :

$$\tan \theta = \frac{r}{z} \quad (2.21)$$

La surface étant spéculaire, nous pouvons écrire :

$$\frac{dz}{dr} = -\tan \frac{\theta}{2} \quad (2.22)$$

$$\tan \theta = \frac{2 \tan \frac{\theta}{2}}{1 - \tan^2 \frac{\theta}{2}} \quad (2.23)$$
$$\frac{-2\frac{dz}{dr}}{1 - \frac{dz}{dr}^2} = \frac{r}{z} \quad (2.24)$$
$$\frac{dz}{dr} = \frac{z}{r} - \sqrt{1 + \left(\frac{r}{z}\right)^2} \quad (2.25)$$

dont la résolution fournit :

$$z = \frac{h^2 - r^2}{2h} \quad (2.26)$$

où h est une constante positive d'intégration. Comme nous l'avons déjà dit, le miroir qui satisfait la contrainte du point de vue unique avec une lentille orthographique est un miroir paraboloidal.

2.2 Cas du miroir hyperboloïdal

Les équations de projection dans le cas du miroir hyperboloïdal sont présentées dans cette section, puisque c'est celui que nous avons utilisé. Le point focal de la caméra est positionné au deuxième foyer de l'hyperbole, pour respecter la contrainte du point de vue unique. Il s'agit d'établir maintenant l'équation de projection d'un point (X, Y, Z) sur le plan image de la caméra. Considérons un repère cartésien

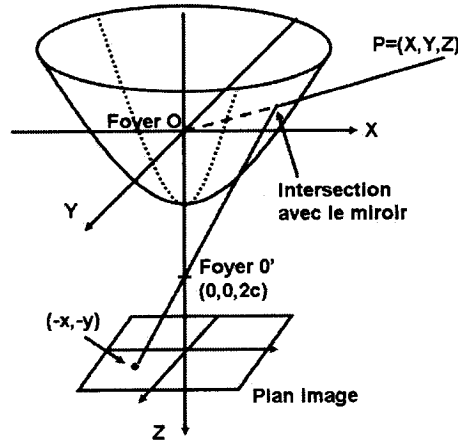


FIG. 2.3 Caméra omnidirectionnelle avec un miroir de forme hyperboloïdale

dont l'origine, le point O , est placée au foyer du miroir, et un point de l'espace

$P = [X, Y, Z]^T$. L'équation du miroir s'écrit alors :

$$\frac{(Z - c)^2}{a^2} - \frac{X^2 + Y^2}{b^2} = 1 \quad (2.27)$$

Le point P est projeté sur le miroir en $P_1 = [X_1, Y_1, Z_1]^T$. Pour calculer les coordonnées de P_1 , considérons la droite passant par P et O d'équation :

$$\nu_1 = \{\mathbf{v} | v = \lambda[X, Y, Z]^T, \lambda \in \mathbb{R}\} \quad (2.28)$$

Le λ se détermine en résolvant l'équation du second degré 2.29 issue de 2.27 :

$$\lambda^2(b^2Z^2 - a^2X^2 - a^2Y^2) - \lambda(2b^2cZ) + b^4 = 0 \quad (2.29)$$

Cette équation fournit le résultat suivant :

$$\lambda = \frac{b^2(cZ \pm a\|\mathbf{X}\|)}{b^2Z^2 - a^2X^2 - a^2Y^2} \quad (2.30)$$

avec

$$\|\mathbf{X}\| = \sqrt{X^2 + Y^2 + Z^2}$$

Le point d'intersection a les coordonnées suivantes : $P_1 = \lambda[X, Y, Z]^T$. Transposons maintenant l'origine du repère au foyer de la deuxième branche de l'hyperbole O' .

Le point P_1 a alors les coordonnées :

$$P_1 = [X'_1, Y'_1, Z'_1]^T = \lambda[X, Y, Z]^T + [0, 0, -2c]^T \quad (2.32)$$

Une projection sur le plan image placé en $Z = -1$ dans le repère centré en O' (la figure 2.3 représente le plan en $Z = 1$, mais stipule bien que les coordonnées

obtenues sont alors $(-x, -y)$ donne P_2 , le point projeté sur le plan image :

$$P_2 = [-X_1/Z_1, -Y_1/Z_1, -1]^T \quad (2.33)$$

Seul le calcul de X_1/Z_1 est ici développé, le calcul de Y_1/Z_1 étant identique :

$$\begin{aligned} X_1/Z_1 &= \frac{\lambda X}{\lambda Z - 2c} \\ X_1/Z_1 &= \frac{1}{Z - 2c \frac{b^2 Z^2 - a^2 X^2 - a^2 Y^2}{b^2 (cZ \pm a\|\mathbf{X}\|)}} X \\ X_1/Z_1 &= \frac{b^2}{Z - \frac{2c}{b^2} (cZ \mp a\|\mathbf{X}\|)} \end{aligned}$$

Soit finalement

$$\begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} = \frac{(c^2 - a^2)}{(c^2 + a^2)Z + 2ac\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2.34)$$

Dans notre cas, les valeurs fournies par le fournisseur sont les suivantes pour l'hyperbole :

- grand axe $a = 28.0950$ mm.
- petit axe $b = 23.4125$ mm.
- Distance entre les deux foyers $2c = 73.1429$ mm

En considérant la caméra utilisée, une webcam avec une focale de $f = 807$ pixels, nous obtenons un cercle omnidirectionnel dans l'image de 316 pixels de rayon, pour un format d'image 640 par 480.

Nous ne développerons pas les autres cas de miroir mais nous allons plutôt placer le cas hyperboloïdal dans un contexte plus général, la famille des caméras omnidirectionnelles.

2.3 Théorie unificatrice

2.3.1 Premières Équations

Souhaitant étudier les applications des caméras omnidirectionnelles sans se soucier de la forme du miroir et en rassemblant tout sous des équations similaires, Daniilidis et Geyer ont développé une théorie unificatrice (Daniilidis, 2000), montrant que toutes les projections sur le plan image des caméras omnidirectionnelles peuvent être remplacées par deux projections successives :

- Tout d’abord une projection sur une sphère unité dont le centre se trouve à une distance m du plan image
- Puis la projection sur le plan image de chaque point sur la sphère à partir d’un centre de projection situé à une distance l du centre de la sphère comme illustré dans la figure 2.4.

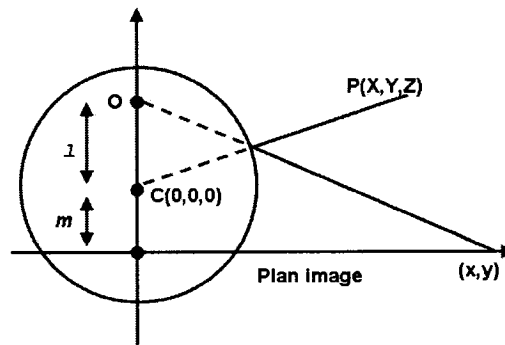


FIG. 2.4 Projections successives

Pour pouvoir être adapter à toutes les caméras omnidirectionnelles, le plan image dessinée à la figure 2.4 devait être placé à une position indépendante des caractéristiques des miroirs mais en un point équivalent pour tous les miroirs. Le foyer de l'hyperboloïde ou du paraboloïde fut donc choisi. Le plan image correspond donc au plan $Z = 0$ de la figure 2.3 avec un repère centré en O . Dans le cadre de la théorie unificatrice, l'équation devient :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{l + m}{Z + l\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix} \quad (2.35)$$

où l et m dépendent des paramètres du miroir utilisé. Cette équation peut s'écrire avec tous les types de miroir qui respectent la contrainte du point de vue unique. Dans notre cas, le miroir est de forme hyperboloïdale et les paramètres s'écrivent de la façon suivante :

$$l = \frac{2ac}{c^2 + a^2} \quad (2.36)$$

$$m = \frac{2c}{c^2 + a^2}(c^2 - a^2 - a) \quad (2.37)$$

Nous retrouvons bien la forme précédente, car :

$$\begin{aligned} l + m &= \frac{2c(c^2 - a^2)}{c^2 + a^2} \\ l &= \frac{2ac}{c^2 + a^2} \end{aligned}$$

Ce qui donne :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{2c(c^2 - a^2)}{(c^2 + a^2)Z + 2ac\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix} \quad (2.38)$$

La théorie unificatrice a été écrite avec une projection en $Z = 0$, c'est-à-dire à une distance de $d = 2c$ du centre de projection, soit le second foyer de l'hyperbole. L'équation est ramenée dans le cas général en divisant par d , et l'équation introduite précédemment est ainsi retrouvée :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{(c^2 - a^2)}{(c^2 + a^2)Z + c\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix}$$

2.3.2 Projection inverse sur la sphère

Cette théorie a l'avantage, en plus de permettre de réunir tous les types de miroir à travers des équations identiques, d'introduire la projection sur la sphère.

Tout d'abord, nous allons exprimer la projection inverse, c'est-à-dire la projection du plan image vers la sphère, en utilisant les paramètres l et m précédemment introduits. Un point sur le plan image à la position (x, y) est projeté sur la sphère au point $(\hat{X}, \hat{Y}, \hat{Z})$ selon l'équation suivante :

$$\begin{aligned} \begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix} &= \frac{l(l+m) + \text{sign}(l+m)\sqrt{(x^2+y^2)(1-l^2) + (l+m)^2}}{x^2+y^2 + (l+m)^2} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.40) \\ \hat{Z} &= \pm\sqrt{1 - \hat{X}^2 - \hat{Y}^2} \end{aligned}$$

avec \hat{Z} qui devient positif si $|l+m|/l > \sqrt{x^2+y^2}$ et négatif sinon. Le cercle de rayon $|l+m|/l$ est appelé «horizon fronto-parallèle». Dans le cas de l'hyperbole, il est visible dans l'image et il représente la limite entre les points de la scène situés en arrière du foyer de l'hyperboloïde et ceux situés en avant du miroir comme l'illustre la figure 2.5.

Pour fournir un ordre de grandeur au lecteur, avec le matériel utilisé (caméra de focale $f = 807$ pixels, et une image de 640 par 480 pixels), l'horizon fronto-parallèle est un cercle de 215 pixels. Pour obtenir la projection inverse de l'équation

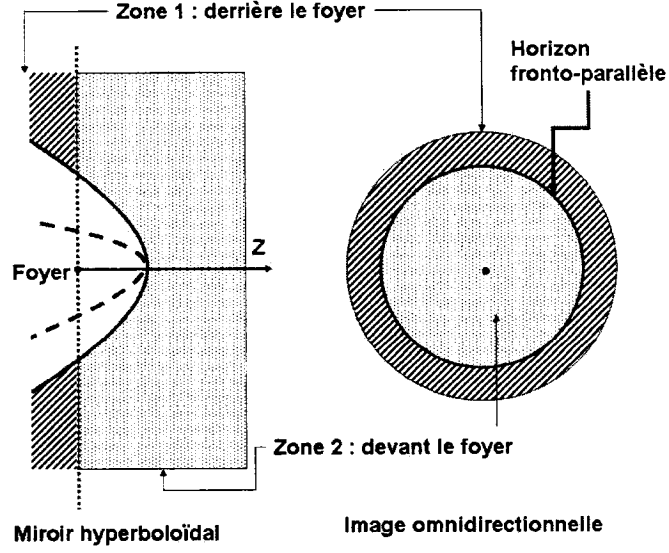


FIG. 2.5 Pour des points situés derrière le foyer de l'hyperbole sur l'axe des Z , leur image est à l'extérieur de l'horizon fronto-parallèle. Pour des points devant le foyer, l'image est à l'intérieur de l'horizon.

2.40, il suffit de reprendre l'équation 2.35. Nous résolvons une équation en $R_{xy} = \sqrt{\hat{X}^2 + \hat{Y}^2}$, en considérant $\hat{X}^2 + \hat{Y}^2 + \hat{Z}^2 = 1$. L'équation est la suivante avec $r = \sqrt{\hat{x}^2 + \hat{y}^2}$:

$$((l + m)^2 + r^2)R_{xy}^2 - 2rl(l + m)R_{xy} + r^2(l^2 - 1) = 0$$

La projection inverse a eu comme intérêt historique premier de transposer des calculs de flux optique du plan image sur la sphère virtuelle ainsi que des calculs de mouvement propre de caméra omnidirectionnelle. Comme il s'agissait de dérivées

de l'espace, le calcul du Jacobien J de cette projection inverse était donc nécessaire :

$$J = \begin{pmatrix} \frac{\partial \hat{X}}{\partial x} & \frac{\partial \hat{X}}{\partial y} \\ \frac{\partial \hat{Y}}{\partial x} & \frac{\partial \hat{Y}}{\partial y} \\ \frac{\partial \hat{Z}}{\partial x} & \frac{\partial \hat{Z}}{\partial y} \end{pmatrix} \quad (2.42)$$

Par contre le Jacobien n'a jamais été calculé dans le cas du miroir hyperboloïdal et c'est pour cela que nous donnons les différents éléments de façon exhaustive ($l + m > 0$, cas du miroir hyperboloïdal) :

$$\begin{aligned} \frac{\partial \hat{X}}{\partial x} &= x^2 \frac{(1-l^2)}{\sqrt{a_1 a_2}} - 2x^2 \frac{a_3}{a_2^2} + \frac{a_3}{a_2} \\ \frac{\partial \hat{X}}{\partial y} &= xy \frac{(1-l^2)}{\sqrt{a_1 a_2}} - 2xy \frac{a_3}{a_2^2} \\ \frac{\partial \hat{Y}}{\partial x} &= xy \frac{(1-l^2)}{\sqrt{a_1 a_2}} - 2xy \frac{a_3}{a_2^2} \\ \frac{\partial \hat{Y}}{\partial y} &= y^2 \frac{(1-l^2)}{\sqrt{a_1 a_2}} - 2y^2 \frac{a_3}{a_2^2} + \frac{a_3}{a_2} \\ \frac{\partial \hat{Z}}{\partial x} &= \text{sign}(\hat{Z}) x \frac{\{[-\frac{(1-l^2)}{\sqrt{a_1}} + 2\frac{a_3}{a_2}](x^2 + y^2) - a_3\} a_3}{a_2^2 \sqrt{1 - \frac{a_3^2}{a_2^2} (x^2 + y^2)}} \\ \frac{\partial \hat{Z}}{\partial y} &= \text{sign}(\hat{Z}) y \frac{\{[-\frac{(1-l^2)}{\sqrt{a_1}} + 2\frac{a_3}{a_2}](x^2 + y^2) - a_3\} a_3}{a_2^2 \sqrt{1 - \frac{a_3^2}{a_2^2} (x^2 + y^2)}} \end{aligned} \quad (2.43)$$

avec

$$\begin{aligned} a_1 &= (x^2 + y^2)(1-l^2) + (l+m)^2 \\ a_2 &= (x^2 + y^2) + (l+m)^2 \\ a_3 &= l(l+m) + \sqrt{A} \end{aligned} \quad (2.44)$$

Le cas du Jacobien pour $l + m < 0$, qui correspond au miroir paraboloidal, est détaillé dans l'annexe II.

2.4 Projection de droites sur le plan image

Dans le prolongement de cette théorie unificatrice, les équations qui décrivent la projection de droites dans l'espace sur le plan image peuvent être établies. Ces calculs ont permis l'établissement de protocoles de calibration dans (Daniilidis, 1999). Ils représentent aussi un intérêt pour le suivi de trajectoire puisqu'ils permettent de connaître la trajectoire, dans l'image, d'objets suivant des lignes rectilignes dans l'espace.

Puisque beaucoup de travaux traitaient le cas de translations perpendiculaires à l'axe de la caméra pour de la robotique mobile (par exemple où la caméra est posé sur un robot, et l'axe perpendiculaire au sol), nous avons prolongé l'étude dans le cas particulier des droites perpendiculaires à l'axe de la caméra. Cette étude permettra par la suite l'expression du flux optique théorique pour une caméra en mouvement, dans le cas de translations perpendiculaires à l'axe de la caméra qui sera développé dans le chapitre 3.

2.4.1 Projection des droites sur la sphère

Toute droite dans l'espace va se projeter sur un grand cercle de la sphère (la projection d'un segment n'est bien sûr qu'un arc de cercle). Un exemple de grand cercle est donné sur la figure 2.6.

2.4.2 Projection des grands cercles sur le plan image

La droite va former avec le centre de la sphère un plan dans l'espace qui va intersecter la sphère et créer ainsi un grand cercle. Le plan est défini par son vecteur

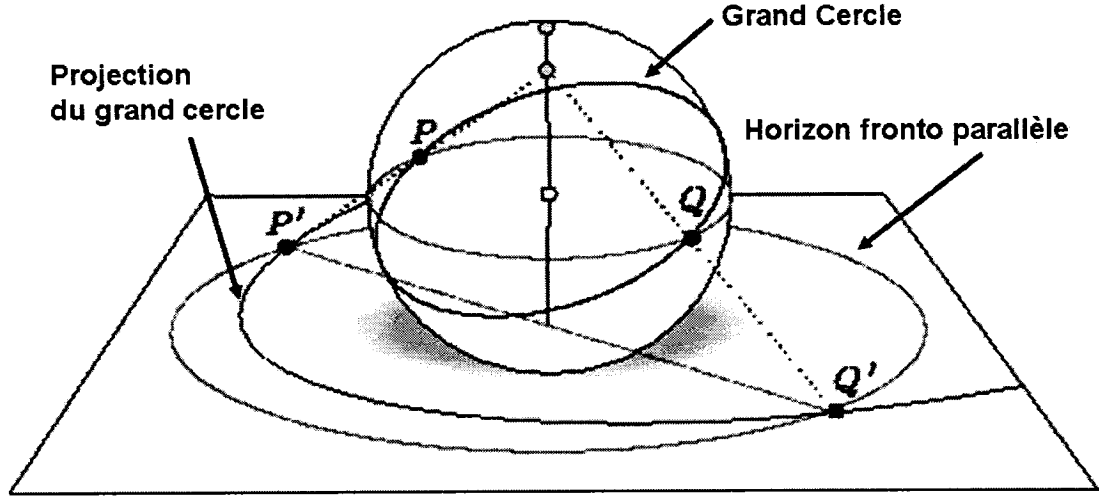


FIG. 2.6 Projections de droites sur le plan image via les deux projections

normal, $\vec{n} = (n_x, n_y, n_z)$. Le cercle dessiné sur la plan image de la figure 2.6 est l'horizon fronto-parallèle et représente la projection du grand cercle parallèle sur le plan image. Par ailleurs un grand cercle quelconque coupe le grand cercle parallèle au plan image en deux points que nous appellerons par la suite P et Q . Les autres grands cercles se projettent suivant des coniques dont les principaux paramètres sont donnés en fonction de \vec{n} .

$$F_i = \left(\frac{(l+m)n_x(n_z \pm \sqrt{1-l^2})}{n_x^2 + n_y^2 - l^2}, \frac{(l+m)n_y(n_z \pm \sqrt{1-l^2})}{n_x^2 + n_y^2 - l^2} \right) \quad (2.45)$$

$$a = \frac{l+m}{\sqrt{l^2 - n_x^2 - n_y^2}} \quad (2.46)$$

$$b = \frac{l(l+m)n_z}{l^2 - n_x^2 - n_y^2} \quad (2.47)$$

où F_i est la position des foyers, a le grand axe et b le petit axe. Ces paramètres sont illustrés sur la figure 2.7.

Toutes les droites comprises dans un même plan coupant le centre de la sphère ont une projection commune sur le plan image, illustrée sur la figure 2.6 (en effet le vecteur \vec{n} est le même pour toutes les droites). Chaque image d'une droite intersecte l'horizon fronto-parallèle en deux points, de façon antipodale, que nous appellerons par la suite P' et Q' .

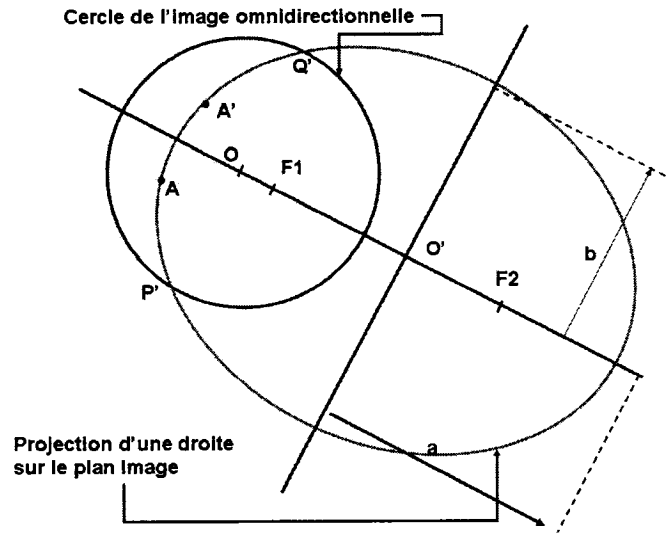


FIG. 2.7 Projection d'une droite sur le plan image : position des foyers et paramètres de la conique.

2.4.3 Cas Particulier : droites parallèles au plan image

Certains de nos calculs ont porté sur un type particulier de droite, celles qui sont parallèles au plan image. Si la caméra effectue une translation \vec{T} dans l'espace, tous les points de l'espace effectuent donc une translation $-\vec{T}$ par rapport à la caméra. Le déplacement de ces points dans l'espace s'appuie donc sur des droites

de direction \vec{T} . Si de plus la caméra effectue une translation perpendiculaire à son axe, toutes les droites sont parallèles au plan image et nous tombons dans le cas particulier que nous allons développer. Il est facile d'obtenir les coordonnées des points P et Q pour une droite de l'espace donnée. Si elle appartient à un plan de l'espace passant par le centre de la sphère de normale $\vec{n} = (n_x, n_y, n_z)$, les coordonnées de P et Q sont :

$$\begin{aligned} x &= \pm \frac{n_y}{n_x} \frac{1}{\sqrt{1 + \frac{n_y^2}{n_x^2}}} \\ y &= \mp \frac{1}{\sqrt{1 + \frac{n_y^2}{n_x^2}}} \\ z &= 0 \end{aligned} \tag{2.48}$$

Nous souhaitons prouver que si deux droites sont parallèles au plan image et qu'elles sont parallèles entre elles alors les points P et Q leur correspondant sont identiques.

Proposition 1 *Les points d'intersection P et Q des grands cercles avec l'horizon fronto-parallèle de deux droites parallèles, elles-mêmes parallèles au plan image, sont identiques.*

Démonstration 1 *Considérons deux droites dans l'espace formées respectivement par les points $A_1 = (a_1, b_1, c_1)$ et $A_2 = (a_2, b_2, c_2)$, et par le vecteur directeur $\vec{t} = (u, v, 0)$. Nous pouvons maintenant écrire les vecteurs \vec{n}_i qui décrivent le*

plan passant par les droites en question et par le centre de la sphère.

$$\begin{aligned} n_{ix} &= \frac{1}{\sqrt{(c_iv)^2 + (c_iu)^2 + (a_iv - b_iu)^2}}(-c_iv) \\ n_{iy} &= \frac{1}{\sqrt{(c_iv)^2 + (c_iu)^2 + (a_iv - b_iu)^2}}(c_iu) \\ n_{iz} &= \frac{1}{\sqrt{(c_iv)^2 + (c_iu)^2 + (a_iv - b_iu)^2}}(a_iv - b_iu) \end{aligned}$$

Or les coordonnées des points P et Q ne dépendent que du rapport $\frac{n_y^2}{n_x^2}$. Ici nous avons $\frac{n_{1y}^2}{n_{2x}^2} = \frac{n_{2y}^2}{n_{2x}^2} = -\frac{u}{v}$. Donc les points P et Q sont les mêmes.,

Par conséquent les points P' et Q' sont identiques pour les droites parallèles au plan image et leurs coordonnées sont facilement calculables :

$$\begin{aligned} x &= \pm \frac{l+m}{l} \frac{n_y}{n_x} \frac{1}{\sqrt{1 + \frac{n_y^2}{n_x^2}}} \\ y &= \mp \frac{l+m}{l} \frac{1}{\sqrt{1 + \frac{n_y^2}{n_x^2}}} \\ z &= -m \end{aligned} \tag{2.49}$$

Nous avons maintenant un point commun à toutes les droites possédant la même direction et qui sont parallèles au plan image : leur projection vont toutes se croiser aux points P' et Q' dans le plan image.

Si nous reprenons l'idée de la translation de la caméra de direction \vec{T} , alors P' et Q' représentent les foyers de contraction et d'expansion dans l'image comme l'illustre la figure 2.8.

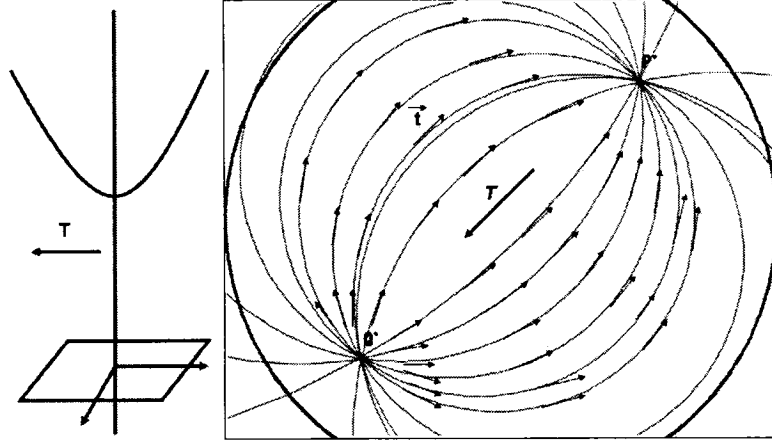


FIG. 2.8 Translation perpendiculaire à l'axe de la caméra : le foyer d'expansion Q' et le foyer de contraction P' sont présents.

Pour résumer ce chapitre, nous pouvons dire que nous avons à notre disposition tous les outils de la géométrie projective des caméras omnidirectionnelles. Nous avons aussi décrit la théorie unificatrice des systèmes catadioptriques, qui nous servira à développer des équations et des protocoles dans le cadre général des caméras omnidirectionnelles. La dernière étape décrit comment déterminer la projection de lignes de l'espace 3D dans le plan image, ce qui nous servira dans le chapitre suivant, pour le calcul du flux optique dans une suite d'images omnidirectionnelles.

CHAPITRE 3

CALCUL DU FLUX OPTIQUE DANS DES IMAGES OMNIDIRECTIONNELLES

Ce chapitre présente l'adaptation d'une technique de calcul de flux optique dans le cas d'images omnidirectionnelles. Tout d'abord les équations générales de calcul du flux optique sont introduites, puis la technique retenue est détaillée. Enfin, l'adaptation de cette technique est développée. La validité du calcul du flux optique est démontrée et une comparaison avec d'autres algorithmes est analysée.

3.1 Introduction au flux optique

Le flux optique se définit comme le mouvement apparent de la luminosité dans une image. Cette introduction présente les premiers calculs et les hypothèses nécessaires au calcul du flux optique dans des images en perspective ainsi que le choix d'une méthode de calcul, la méthode différentielle de Lucas et Kanade (Lucas, 1981).

3.1.1 Premiers calculs

La première hypothèse à considérer pour les calculs est la suivante :

- l'intensité dans l'image est une fonction du temps et de la position sur le plan image $(x,y) : I(x,y,t)$

Soit un point P de l'espace 3D visible par la caméra : sa projection sur le plan image est en (x,y) à l'instant t et en $(x + dx, y + dy)$ à l'instant $t + dt$. L'intensité de l'image correspondant à la projection du point P sur le plan image est :

- $I(x, y, t)$ à l'instant t au point (x, y) ,
- $I(x + dx, y + dy, t + dt)$ à l'instant $t + dt$ au point $(x + dx, y + dy)$.

Avec la première hypothèse, les intensités $I(x, y, t)$ et $I(x + dx, y + dy, t + dt)$ peuvent être reliées ainsi :

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \sum_{n=1}^{+\infty} \frac{1}{n!} (dx \frac{\partial I}{\partial x}(x, y, t) + dy \frac{\partial I}{\partial y}(x, y, t) + dt \frac{\partial I}{\partial t}(x, y, t))^n \quad (3.1)$$

L'hypothèse de calcul la plus importante est la suivante :

- Malgré l'intervalle de temps et les changements qui peuvent se produire (Ex :le point P se déplace, la caméra se déplace), l'intensité correspondant au point P est la même dans les deux images, c'est-à-dire en t et en $t + dt$.

Cela se traduit par

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (3.2)$$

En négligeant les termes d'ordre supérieur, l'équation 3.1 devient :

$$(dx \frac{\partial}{\partial x} + dy \frac{\partial}{\partial y} + dt \frac{\partial}{\partial t}) I(x, y, t) \simeq 0 \quad (3.3)$$

Écrit plus simplement, cela donne l'expression appelée *équation de contrainte du flux optique* :

$$\frac{\partial I}{\partial x}(x, y, t) \cdot u + \frac{\partial I}{\partial y}(x, y, t) \cdot v + \frac{\partial I}{\partial t}(x, y, t) \simeq 0 \quad (3.4)$$

où (u, v) sont les composantes du flux optique à calculer.

3.1.2 Méthode différentielle avec hypothèse de constance locale

L'équation 3.4 est à deux inconnues et les différentes techniques existantes consistent à prendre une hypothèse supplémentaire pour la résoudre.

La technique différentielle de B.Lucas et T.Kanade (Lucas, 1981) consiste à supposer que la valeur (u, v) est la même sur un voisinage Ω autour du point (x, y) .

Cette méthode que nous nommerons dans la suite du mémoire, *méthode différentielle avec constance locale* se décompose en plusieurs étapes :

1. pour chaque point dans le voisinage Ω , il faut écrire l'équation 3.4 correspondante,
2. construire le système d'équations correspondant avec autant d'équation que de points dans le voisinage,
3. résoudre le système par minimisation des moindres carrés.

Il s'agit de minimiser l'énergie E égale à la norme L_2 de la différence $I(x + dx, y + dy, t + dt) - I(x, y, t)$, soit :

$$E = \sum_{(x,y) \in \Omega} \left(\frac{\partial I}{\partial x}(x, y, t) \cdot u + \frac{\partial I}{\partial y}(x, y, t) \cdot v + \frac{\partial I}{\partial t}(x, y, t) \right)^2 \quad (3.5)$$

En introduisant une fonction de poids $W(x, y)$, l'énergie à minimiser devient finalement :

$$E = \sum_{(x,y) \in \Omega} W(x, y)^2 \left(\frac{\partial I}{\partial x}(x, y, t) \cdot u + \frac{\partial I}{\partial y}(x, y, t) \cdot v + \frac{\partial I}{\partial t}(x, y, t) \right)^2 \quad (3.6)$$

Dans le cas d'un voisinage contenant n points (x_i, y_i) pour $i = 1, \dots, n$, l'équation 3.6 devient :

$$E = \sum_{i=1}^n W(x_i, y_i)^2 \left(\frac{\partial I}{\partial x}(x_i, y_i, t) \cdot u + \frac{\partial I}{\partial y}(x_i, y_i, t) \cdot v + \frac{\partial I}{\partial t}(x_i, y_i, t) \right)^2 \quad (3.7)$$

la solution est donnée par l'expression :

$$A^T W^2 A \begin{pmatrix} u \\ v \end{pmatrix} = A^T W^2 \mathbf{b} \quad (3.8)$$

dans laquelle :

$$\begin{aligned} A &= \begin{pmatrix} \frac{\partial I}{\partial x}(x_1, y_1, t) & \dots & \frac{\partial I}{\partial x}(x_n, y_n, t) \\ \frac{\partial I}{\partial y}(x_1, y_1, t) & \dots & \frac{\partial I}{\partial y}(x_n, y_n, t) \end{pmatrix}^T \\ W &= \text{diag}[W(x_1, y_1), \dots, W(x_n, y_n)] \\ b &= -[\frac{\partial I}{\partial t}(x_1, y_1, t), \dots, \frac{\partial I}{\partial t}(x_n, y_n, t)] \end{aligned}$$

Les différents termes s'écrivent alors :

$$A^T W^2 A = \begin{pmatrix} \sum_{i=1}^n W^2(x_i, y_i) I_x^2(x_i, y_i, t) & \sum_{i=1}^n W^2(x_i, y_i) I_x(x_i, y_i, t) I_y(x_i, y_i, t) \\ \sum_{i=1}^n W^2(x_i, y_i) I_x(x_i, y_i, t) I_y(x_i, y_i, t) & \sum_{i=1}^n W^2(x_i, y_i) I_y^2(x_i, y_i, t) \end{pmatrix} \quad (3.9)$$

et

$$A^T W^2 \mathbf{b} = \begin{pmatrix} \sum_{i=1}^n W^2(x_i, y_i) I_x(x_i, y_i, t) I_t(x_i, y_i, t) \\ \sum_{i=1}^n W^2(x_i, y_i) I_y(x_i, y_i, t) I_t(x_i, y_i, t) \end{pmatrix} \quad (3.10)$$

avec

$$\begin{aligned} I_x(x, y, t) &= \frac{\partial I}{\partial x}(x, y, t) \\ I_y(x, y, t) &= \frac{\partial I}{\partial y}(x, y, t) \\ I_t(x, y, t) &= \frac{\partial I}{\partial t}(x, y, t) \end{aligned}$$

La solution est alors $(u, v) = [A^T W^2 A]^{-1} A^T W^2 \mathbf{b}$ si la matrice $A^T W^2 A$ est inversible. Quand la matrice n'est pas inversible, les codages généralement effectués consistent à calculer le flux normal donné par :

$$(u, v) = \frac{-I_t(x, y, t)}{\|\nabla I(x, y, t)\|} \cdot \frac{\nabla I(x, y, t)}{\|\nabla I(x, y, t)\|} \quad (3.11)$$

Les références bibliographiques (Heeger, 1991), (Barron, 1994) et (Daniilidis, 2002) utilisent des voisinages gaussiens pour la matrice W . Ils pensent qu'il faut accorder plus d'importance au point (x, y) que pour les points en périphérie du voisinage. Cependant, prendre des poids identiques de part et d'autre du point central (x, y) suppose que l'égalité des distances dans l'image implique l'égalité des distances dans la scène, ce qui est inexact en général. De plus, garder les mêmes poids d'une position de l'image à une autre dans l'image est inexact pour la même raison. Cependant, toutes les mises en pratique des articles (Heeger, 1991), (Barron, 1994) et (Daniilidis, 2002) dans le cas d'images en perspective utilisent quand même un voisinage gaussien centré sur le point d'intérêt et identique en tout point de l'image.

Le choix de la méthode différentielle à constance locale en particulier pour l'adaptation au cas des images omnidirectionnelles a été motivé par deux raisons :

- Dans (Stratmann, 2002), une caméra est posée sur un robot qui subit soit des translations pures soit des trajectoires curvilignes. Cela permet d’obtenir plusieurs séquences d’images. Plusieurs techniques de calculs de flux optique sont testées sur ces images omnidirectionnelles dans le but d’estimer la position des foyers de contraction et d’expansion. Ces foyers permettent d’estimer les directions de déplacement du robot. La technique différentielle décrite précédemment possède le meilleur compromis entre temps de calcul et précision.
- Comme cité dans la section 1.1.1.3, (Daniilidis, 2002) est le seul article qui propose une réelle adaptation d’une méthode de calcul de flux optique et il utilise la méthode différentielle précédemment exposée.

3.2 Opérateurs appliqués sur des images omnidirectionnelles

A partir de maintenant, comme nous allons aborder le cas des images omnidirectionnelles, et que celles-ci possèdent des propriétés de symétrie par rapport à leur centre, nous utiliserons le terme *centre de l’image* pour parler du point dans l’image qui correspond à l’image du centre du miroir et la position d’un pixel pourra être définie par son *rayon*, distance du pixel au centre de l’image. En effet, si la caméra est bien calibrée, le contour d’une image omnidirectionnelle est un cercle et le centre de l’image est le centre du cercle. Nous appellerons *périphérie de l’image* la région de l’image qui correspond à l’image de la périphérie du miroir.

3.2.1 Problèmes avec les opérateurs habituels

Comme expliqué dans la section 1.1.1.3, le recours à des opérateurs habituellement utilisés pour des images en perspective pose problème. Pour une image en perspective, l’utilisation d’opérateurs homogènes dans toute l’image, c’est-à-dire identiques

quel que soit le point sur lequel ils sont centrés, n'est pas justifiée mais est utilisée couramment. Dans des images omnidirectionnelles, de tels opérateurs ne reflètent pas non plus les déformations locales présentes dans l'image. Dans l'annexe I, il est montré que la résolution diminue en s'éloignant du centre de l'image. Le lecteur peut s'en rendre compte en considérant une portion d'image de taille fixe au centre de l'image et la même portion à la périphérie. Celle du centre correspond à une portion de l'espace moins grande que celle de la périphérie comme l'illustre la figure 3.1 pour une coupe de la caméra omnidirectionnelle. La portion est ici représentée par un intervalle. Si des distances égales en pixels dans l'image correspondent à des angles différents dans l'espace, ces différences d'angles varient avec la position radiale du voisinage dans l'image. Sur la figure 3.2, les points a_1 et a_2 sont chacun aussi éloignés de a_0 , mais ils n'ont pas le même rayon (distance au centre de l'image). Comme pour les intervalles de la figure 3.1, si nous imaginons un opérateur centré en a_0 , la pondération en a_1 et a_2 doit refléter que les portions de l'espace projetées en $[a_0a_1]$ et en $[a_0a_2]$ sont de taille différente. L'utilisation d'opérateurs homogènes dans l'image doit donc être évitée.

3.2.2 Transposer le calcul sur la sphère

Dans la section 1.1.1.3, nous avons mentionné la technique choisie par (Daniilidis, 2002) pour calculer le flux optique. Il s'agit de revenir dans un espace où les opérateurs sont encore valables. Il définit les opérateurs mathématiques nécessaires à la transposition du calcul sur la sphère décrite (projection inverse et Jacobien). Un opérateur gaussien sur la sphère est ensuite défini et lorsqu'il est centré en un point précis de la sphère, il sert à définir le voisinage pour le calcul du flux. Il est ici homogène sur toute la sphère, puisqu'il présente la même distribution quelque soit le point où il est centré. C'est sur ce voisinage que le flux est supposé constant pour la minimisation des moindres carrés décrite par l'équation 3.6. L'équation du

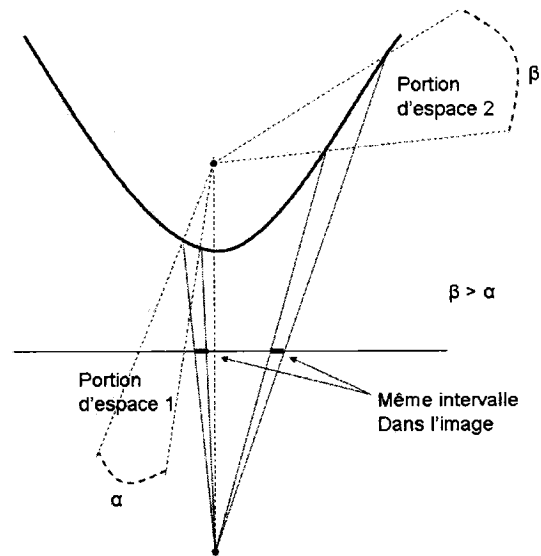


FIG. 3.1 Deux intervalles de même taille ne correspondent pas à une portion d'espace de même taille.

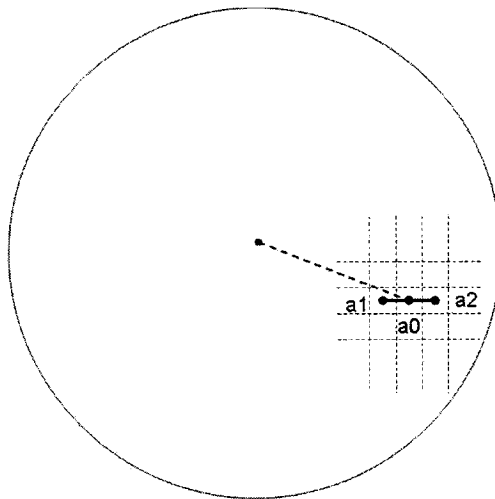


FIG. 3.2 a_1 et a_2 sont chacun aussi éloigné de a_0 , mais la portion de l'espace projeté dans le segment a_1a_0 est plus petite que celle projetée en a_0a_2

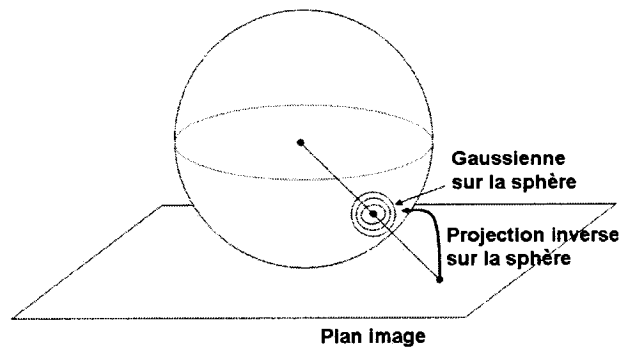


FIG. 3.3 Le calcul est transposé du plan image sur la sphère. Les opérateurs sont exprimés sur la sphère.

flux optique 3.4 est transposée en coordonnées sphériques, mais la résolution reste identique à celle décrite en section 3.1.2.

Nous avons souhaité garder les calculs sur le plan image en conservant l'équation du flux 3.4 dans le système de coordonnées du plan image. Au lieu de transposer le calcul sur un autre ensemble, nous avons transposé les opérateurs nécessaires au calcul de flux sur le plan image. La section suivante décrit la méthode de calcul des opérateurs en fonction de leur position dans l'image.

3.2.3 Opérateurs à adapter

En observant la méthode de calcul de flux deux sortes d'opérateurs vont nécessiter une adaptation :

1. L'opérateur qui va définir le voisinage Ω pour la minimisation des moindres carrés (équation 3.6)

2. Les opérateurs de dérivation spatiale par rapport à x et à y .

Comme nous conservons les calculs sur le plan image, l'objet de cette section est d'exprimer la méthode de calcul des opérateurs dans le plan image que nous allons utiliser. Par la suite, nous utiliserons le terme *taille d'un opérateur* pour parler du nombre de composantes matricielles non nulles. De façon qualitative, ce la représente le nombre de pixels sur lequel il agit véritablement.

En ce qui concerne le voisinage à utiliser (matrice W de la section 3.1.2), des ouvrages de référence recommandent l'opérateur gaussien qui, lorsqu'il est centré en (X_0, Y_0) et de variance σ , s'exprime ainsi :

$$G_{(X_0, Y_0)}(X, Y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}((X-X_0)^2 + (Y-Y_0)^2)} \quad (3.12)$$

Toutes les applications pratiques observées de cet opérateur dans le cadre du flux optique utilisent une approximation, avec des valeurs finies, de cet opérateur. Par exemple, pour un voisinage de 3 par 3 pixels, une des approximations d'un opérateur gaussien de grande variance est :

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (3.13)$$

Dans la section précédente, nous avons vu que l'opérateur devait s'adapter à sa position dans l'image. Comme la résolution de l'image diminue en s'éloignant du centre, nous pouvons déjà penser intuitivement que la taille doit diminuer en s'éloignant du centre de l'image. Il faut donc d'abord obtenir des approximations des opérateurs de taille supérieure à 3 par 3. En ce qui concerne l'opérateur gaussien, il est facile d'obtenir des approximations, avec des valeurs finies, de n'importe quelle taille souhaitée.

Par contre, c'est un peu plus complexe avec les opérateurs de dérivations. Les plus couramment utilisés sont les opérateurs de Sobel qui fournissent en un point de l'image les dérivées en x et y :

$$s1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad s2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (3.14)$$

Pour obtenir ces opérateurs de Sobel, il faut faire la convolution entre une approximation d'une gaussienne à une dimension et l'opérateur de dérivation adéquat à chaque fois.

Pour la dérivée selon x :

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.15)$$

Pour la dérivée selon y :

$$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.16)$$

Il est conseillé d'utiliser une gaussienne en deux dimensions pour la convolution avec les deux opérateurs de dérivation *de base*, DH et DV :

$$DH = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \quad DV = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}^T \quad (3.17)$$

Nous avons fait pareil pour obtenir des opérateurs de dérivation de taille supérieure à 3 par 3. Par exemple, pour un opérateur de dérivation de taille 5 par 5, nous

prenons une approximation de l'opérateur gaussien de 3 par 3, et nous faisons la convolution avec DH et DV :

$$\begin{pmatrix} -1 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ -1 & -2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.18)$$

Pour résumer, nous disposons maintenant des opérateurs nécessaires au calcul au calcul de flux optique, et cela pour différentes tailles. Il reste à les adapter pour les images omnidirectionnelles.

3.2.4 Adaptation des opérateurs

Dans le cas d'une image en perspective, pour un support donné dans le plan image, un cône spatial peut être dessiné avec comme sommet le point de vue de la caméra. Ce cône représente la partie de l'espace vue par ce support, comme le montre le dessin de gauche de la figure 3.4. C'est par exemple sur ce support que va agir un opérateur. Pour un même cône spatial, nous avons vu que même dans le cas d'image en perspective la taille de l'opérateur devrait évoluer en fonction de sa position dans l'image. Ce n'est pas le cas dans la plupart des applications pratiques, dans lesquelles les opérateurs sont identiques en tout point de l'image. Mais nous ne sommes pas dans le cas de la partie gauche de la figure 3.4, nous avons en plus une réflexion sur le miroir à considérer. Nous pouvons néanmoins utiliser un raisonnement similaire.

Nous avons vu qu'une caméra omnidirectionnelle possédait un point de vue effectif,

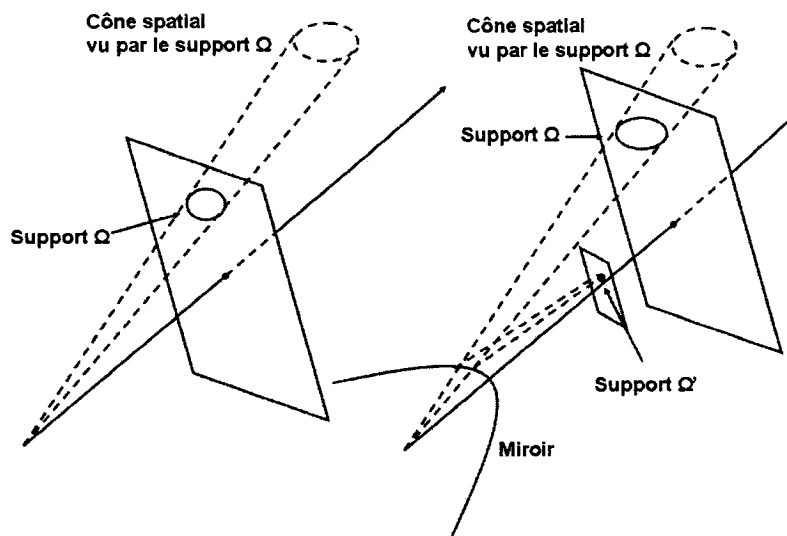


FIG. 3.4 Sans miroir : Cône spatial vu par un support donné Ω du plan image. Avec miroir : même cône spatial vu par le point de vue «effectif» de la caméra omnidirectionnelle.

que nous pouvons substituer au point de vue de la caméra en perspective pour le besoin de notre démonstration (partie de droite de la figure 3.4).

Dans la pratique les opérateurs sont pris identiques en tout point d'un plan perpendiculaire à l'axe de la caméra, appelé plan image. Notre idée est de reprendre cette constance sur un plan perpendiculaire à l'axe de la caméra, mais de faire subir à un opérateur présent sur ce plan l'effet de la projection sur le miroir.

Sur la partie de droite de la figure 3.4, nous pouvons voir l'opérateur homogène présent sur un plan $Z = Z_0$. Il est projeté sur le plan image de la caméra omnidirectionnelle après réflexion sur le miroir.

Pour vraiment respecter la géométrie du miroir, l'idée intuitive était de faire *subir* aux opérateurs l'effet de la projection sur le plan image que subissent les points de l'espace. Nous sommes donc partis de l'idée des opérateurs homogènes sur un plan parallèle au plan image, pour les projeter sur le plan image. Si nous prenons le système de coordonnées de la figure 3.5, il s'agit d'un plan $Z = Z_0$. En considérant

un opérateur homogène sur ce plan, plus il est centré en un point éloigné de l'axe des Z , plus son image sur le plan image diminue, tout en respectant les déformations de forme liées au système catadioptrique.

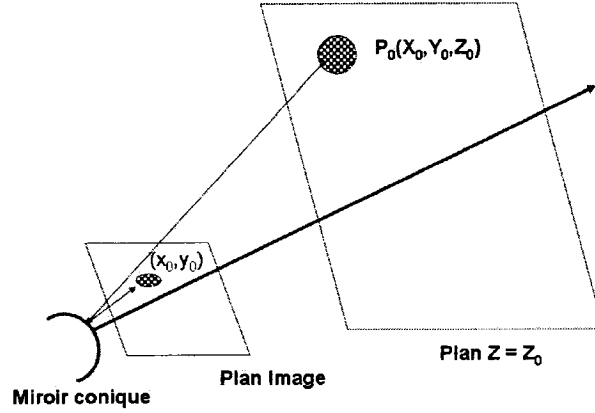


FIG. 3.5 Projection de l'opérateur d'un plan $Z = Z_0$ sur le plan image avec une réflexion sur le miroir.

Dans la pratique, il est plus facile de procéder de façon inverse. Nous cherchons pour tout point (x, y) du plan image, la valeur d'un opérateur centré en n'importe quel point (x_0, y_0) . En calculant la projection inverse des points (x, y) et (x_0, y_0) sur le plan $Z = Z_0$, nous obtenons respectivement P et P_0 . À partir de ces deux points, il est possible de calculer la valeur en P , de l'opérateur centré en P_0 , en utilisant la formule 3.12. Cette valeur est celle de notre opérateur adapté au point (x, y) , centré en (x_0, y_0) . Nous appellerons par la suite adaptation des opérateurs, l'action d'obtenir ces opérateurs par projection de l'opérateur de ce plan $Z = Z_0$ sur le plan image avec réflexion sur le miroir.

Nous allons donner les formules pour la projection inverse sur le plan $Z = Z_0$ avant

de donner les résultats de notre adaptation pour la matrice W de voisinage et pour les opérateurs de dérivation.

3.2.4.1 Projection inverse sur un plan $Z = Z_0$

Grâce à la projection inverse sur la sphère décrite par l'équation 2.40, les coordonnées du point $(\hat{X}, \hat{Y}, \hat{Z})$, projeté inverse de (x, y) , sont données par les formules :

$$\begin{aligned} \begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix} &= \frac{l(l+m) + \text{sign}(l+m)\sqrt{(x^2+y^2)(1-l^2) + (l+m)^2}}{x^2 + y^2 + (l+m)^2} \begin{pmatrix} x \\ y \end{pmatrix} \\ \hat{Z} &= \pm \sqrt{1 - \hat{X}^2 - \hat{Y}^2} \end{aligned}$$

où \hat{Z} est positif si $|l+m|/l > \sqrt{x^2+y^2}$ et négatif sinon. Le point (X, Y, Z_0) , dont le projeté sur la sphère est $(\hat{X}, \hat{Y}, \hat{Z})$ est donné par la formule suivante :

$$\begin{pmatrix} X \\ Y \\ Z_0 \end{pmatrix} = \sqrt{X^2 + Y^2 + Z_0^2} \begin{pmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{pmatrix} \quad (3.19)$$

Finalement, la projection inverse du point (x, y) sur le plan $Z = Z_0$ s'exprime ainsi :

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \frac{Z_0}{\pm \sqrt{1 - \hat{X}^2 - \hat{Y}^2}} \cdot \frac{l(l+m) + \text{sign}(l+m)\sqrt{(x^2+y^2)(1-l^2) + (l+m)^2}}{x^2 + y^2 + (l+m)^2} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.20)$$

Pour plus de simplicité nous noterons par la suite

$$(X, Y) = f(x, y) = (f_1(x, y), f_2(x, y)) \quad (3.21)$$

3.2.4.2 Voisinage

L'opérateur gaussien «adapté» G' , au point (x, y) , centré en tout point de l'image (x_0, y_0) est maintenant égal à :

$$\begin{aligned} G'_{(x_0, y_0)}(x, y) &= G_{f(x_0, y_0)} * f(x, y) \\ G'_{(x_0, y_0)}(x, y) &= \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}((f_1(x, y) - f_1(x_0, y_0))^2 + (f_2(x, y) - f_2(x_0, y_0))^2)} \end{aligned} \quad (3.22)$$

Il est maintenant possible de remplacer l'opérateur $W(x, y)$, utilisé pour le voisinage Ω lors du calcul du flux optique (voir section 3.1.2), par $G'_{(x_0, y_0)}(x, y)$ en tout point (x_0, y_0) de l'image.

Dans la suite, la notation $G'(i_0, j_0)$ sera utilisée pour désigner la matrice qui approxime l'opérateur $G'_{(x_0, y_0)}$ au pixel (i_0, j_0) pendant le calcul avec des valeurs entières. L'évolution d'un noyau gaussien centré de plus en plus loin du centre de l'image, est illustrée à la figure 3.6.

Dans la pratique, il fallait trouver une taille maximale pour notre voisinage. Ce voisinage dit initial, sera projeté sur le plan image par la méthode décrite dans la section 3.2.4. Nous avons donc créé plusieurs voisinages de taille importante (11 par 11 pixels, 13 par 13 pixels, 15 par 15 pixels), et nous avons testé leur évolution du centre à la périphérie de l'image en utilisant la méthode d'adaptation précédemment décrite. Le voisinage 11 par 11 au centre de l'image a été choisi et

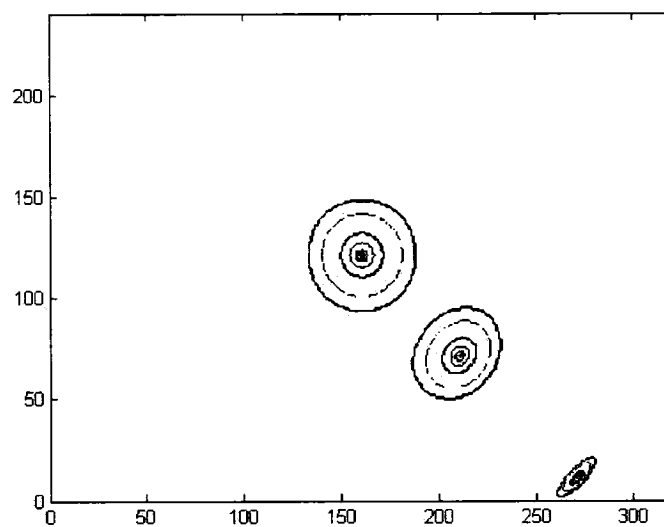


FIG. 3.6 Évolution du voisinage en s'éloignant du centre.

décroît en taille jusqu'à une taille minimale de 3 par 3 que nous avons fixé aussi pour qu'il garde une signification mathématique comme opérateur dans une image.

Pour un point à la position $[100, 100]$:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 252 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1200 & 14400 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1200 & 63504 & 1200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 252 & 14400 & 1200 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.25)$$

3.2.4.3 Opérateurs de dérivation

Pour obtenir les opérateurs de dérivation, nous avons d'abord calculé les opérateurs de dérivation de taille 11 par 11 en reprenant l'idée de la section 3.2.3. Il faut pour cela faire la convolution d'une approximation d'une gaussienne de taille 9 par 9 avec les matrices DH et DV :

$$DH = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \quad DV = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}^T \quad (3.26)$$

Ensuite, nous avons repris la technique décrite dans la section 3.2.4, en projetant les opérateurs sur la plan image.

Pour un point à la position $[100, 100]$:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -28 & -784 & 28 & 784 & 0 & 0 & 0 \\ 0 & 0 & 0 & -28 & -4900 & 0 & 4900 & 28 & 0 & 0 & 0 \\ 0 & 0 & 0 & -784 & -28 & 784 & 28 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.29)$$

Ainsi, nous détenons tous les outils nécessaires pour notre algorithme de flux optique et nous allons le tester. Pour plus de simplicité, nous appellerons par la suite les *opérateurs adaptés* ceux obtenus par cette méthode et les *opérateurs homogènes* ceux obtenus par la méthode de la section 3.2.3.

3.3 Méthode d'analyse de l'algorithme de flux optique

3.3.1 Choix d'une variable pour analyser l'effet des nouveaux opérateurs sur le calcul du flux optique

Maintenant que l'adaptation des opérateurs dans le cas omnidirectionnel a été réalisée, il faut valider notre calcul de flux et comparer cette adaptation à un calcul de flux optique sans adaptation. Il faut séparer les deux paramètres majeurs d'un test de validation de flux optique : la norme et l'orientation du flux. En ce

qui concerne la norme, beaucoup d'équipes ont recours à des images synthétiques, car il est nécessaire d'avoir des environnements et des déplacements calibrés pour pouvoir estimer l'amplitude théorique du flux. Pour des séquences d'images réelles, il reste l'étude de l'orientation du flux optique comme le suggèrent deux articles majeurs dans l'étude des performances des techniques de calcul de flux, (Barron, 1994) et (McCane, 2001). Deux paramètres peuvent être analysés :

1. La direction angulaire du flux optique en tout point, avec $\psi_E = \arccos \vec{v}_c \cdot \vec{v}_e$ où \vec{v}_c est le vrai vecteur et \vec{v}_e le vecteur trouvé.
2. Dans (Daniilidis, 2002), la position angulaire du foyer d'expansion est utilisée pour juger de la fiabilité du calcul de flux. Lors d'un déplacement de la caméra omnidirectionnelle de façon perpendiculaire à l'axe optique, le vecteur pointant sur le foyer d'expansion est estimé et est comparé au vecteur attendu.

C'est la première solution qui a été choisie ; elle est illustrée à la figure 3.7.

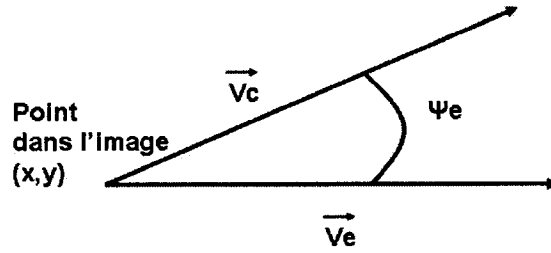


FIG. 3.7 Erreur angulaire entre le vecteur flux théorique et le vecteur estimé.

3.3.2 Flux optiques calculés

Pour compléter l'étude, plusieurs mises en oeuvre de la technique différentielle avec constance locale ont été programmées avec différentes tailles d'opérateurs. Le calcul de flux optique a donc été programmé par la même technique différentielle en

utilisant des opérateurs de taille 3 par 3, 5 par 5, 7 par 7, 9 par 9 et 11 par 11. Ces opérateurs ne sont pas adaptés au cas omnidirectionnels, mais sont là pour comparer le programme qui adapte les opérateurs avec des mises en oeuvre traditionnelles de la technique différentielle avec constance locale pour différentes tailles d'opérateurs. Ces opérateurs homogènes sont obtenus par la méthode explicitée dans la section 3.2.3.

Pour apprécier l'efficacité des algorithmes, le premier outil choisi est l'erreur moyenne commise en orientation, c'est-à-dire le calcul de l'erreur en orientation (différence entre orientation obtenue et orientation théorique) pour chaque pixel, moyennée sur toute l'image. Ce calcul fournit un premier indicatif sur la validité du calcul mais utilisé seul, il n'est pas suffisant. Le deuxième outil pour l'analyse du flux optique est le tracé des diagrammes d'erreur cumulative. Ces diagrammes présentent pour une image I , le pourcentage de pixels $f(x)$ dont l'erreur est inférieure à un certains taux x :

$$f(x) = \frac{\|P\|}{\|I\|} \quad (3.30)$$

avec

$$P = \{p \in I / \text{Erreur}(p) \leq x\} \quad (3.31)$$

Dans les diagrammes présentés plus loin, «OmniFlux» désignera l'algorithme avec les opérateurs adaptés. Pour les algorithmes avec des opérateurs homogènes, ils seront désignés par la taille des opérateurs. Par exemple, si nous parlons de l'Opérateur 3x3, il s'agira pour les opérateurs de dérivation, à un facteur de normalisation près, de :

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (3.32)$$

La matrice W de la section 3.1.2 sera quant à elle

$$W = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (3.33)$$

3.4 Flux optique théorique

Pour comparer les calculs de flux réalisés avec ou sans adaptation, il est nécessaire de savoir quelle est l'orientation théorique du flux dans les images utilisées, puisque c'est cette valeur qui va être utilisée, non seulement pour valider notre calcul de flux optique, mais aussi pour comparer notre méthode avec d'autres. Trois cas distincts ont été choisis et sont décrits dans la suite.

3.4.1 Rotation autour de l'axe de la caméra

Dans le cas d'une rotation autour de l'axe de la caméra, l'orientation du flux est simple à calculer puisque les vecteurs du flux sont tangents à des cercles concentriques centrés de même centre que l'image omnidirectionnelle. Le sens dépend alors du sens de rotation de la caméra comme illustré à la figure 3.8.

En (x, y) , le vecteur unitaire qui a le même sens et la même direction que le flux optique est :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{\sqrt{x^2 + y^2}} \begin{pmatrix} -\epsilon y \\ \epsilon x \end{pmatrix} \quad (3.34)$$

avec $\epsilon = -1$ si R est dans le sens positif, et $\epsilon = 1$ sinon.

Si l'orientation pour chaque pixel est calculé, la moyenne sur toute l'image doit être nulle.

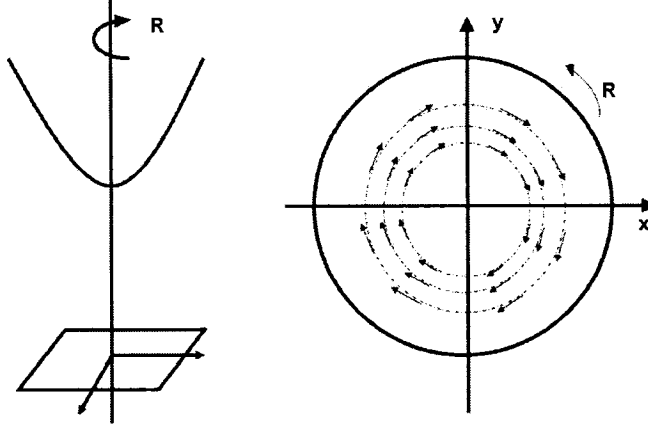


FIG. 3.8 Direction et sens du flux optique pour une rotation R autour de l'axe de la caméra.

3.4.2 Translation dans la direction de l'axe de la caméra

Dans le cas d'une translation dans la direction de l'axe de la caméra, le calcul est également simple puisqu'il ressemble au cas d'une caméra en perspective. Le flux optique s'appuie sur des lignes partant du centre de l'image, qui a selon le sens de translation le rôle d'un foyer d'expansion ou de contraction. Ces mouvements sont illustrés à la figure 3.9.

En (x, y) , le vecteur unitaire qui a le même sens et la même direction que le flux optique est :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{\sqrt{x^2 + y^2}} \begin{pmatrix} \epsilon x \\ \epsilon y \end{pmatrix} \quad (3.35)$$

avec $\epsilon = \pm 1$ en fonction du sens de translation de la caméra.

Si l'orientation pour chaque pixel est calculé, la moyenne sur toute l'image doit être nulle.

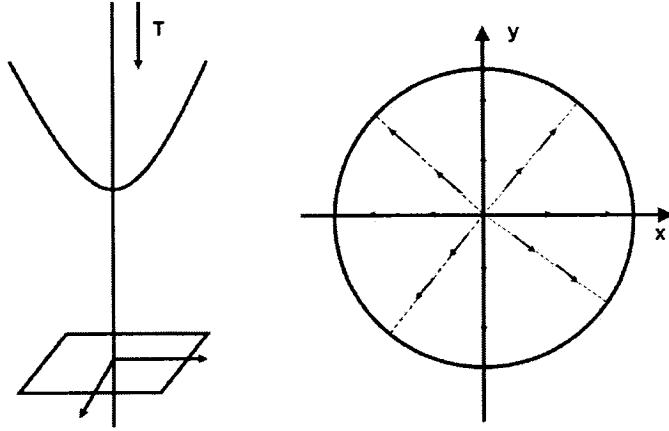


FIG. 3.9 Direction et sens du flux optique pour une translation dans l'axe de la caméra.

3.4.3 Translation perpendiculaire à l'axe de la caméra

Le cas de la translation perpendiculaire à l'axe de la caméra est un peu plus complexe et fait appel à la géométrie projective définie au chapitre 2.

Comme pour les deux types de mouvement précédent, il s'agit de trouver les courbes C dont les vecteurs de flux optique sont tangents. Si la caméra effectue une translation \vec{T} dans l'espace, tous les points de l'espace effectuent une translation $-\vec{T}$ par rapport à la caméra. Le déplacement de ces points dans l'espace s'appuie donc sur des droites de direction \vec{T} . Le déplacement des projections des points de l'espace sur le plan image va donc suivre la trajectoire suivie par la projection de ces droites. Or nous avons étudié la projection des droites de l'espace dans l'image omnidirectionnelle, et plus précisément dans le cas des droites perpendiculaires à l'axe de la caméra. Ce cas nous intéresse particulièrement car il correspond aux translations perpendiculaires à l'axe de la caméra.

Nous allons maintenant nous référer au travail effectué au chapitre 2, en utilisant les projections sphériques de la théorie unificatrice et tout particulièrement à la section 2.4.3. Le lecteur pourra se référer à la figure 3.10 qui illustre le calcul suivant.

Nous avons vu que nous pouvions calculer la projection de n'importe quelle droite de l'espace sur le plan image, et nous savons que c'est une ellipse. La difficulté à faire le chemin inverse : au lieu de chercher la projection d'une droite de l'espace, nous cherchons pour tout point de l'image, l'ellipse dont il fait partie, qui est elle-même la projection dans le plan image d'une droite de l'espace 3D.

Pour calculer les paramètres de cette ellipse, et en reprenant les notations de la section 2.4.3, il suffit déterminer le vecteur \vec{n} , normal au plan formé par les points A , P et Q (A est le projeté inverse de A' sur la sphère comme le montre la figure 3.10). Grâce aux composantes de ce vecteur, les paramètres de l'ellipse sont calculables (le détail des calculs est donné dans le système 2.48).

Pour calculer les coordonnées des points A , P et Q , il faut déterminer d'abord leur projetés sur le plan image A' , P' et Q' (nous dirons aussi que A , P et Q sont les projetés inverses de A' , P' et Q').

Les points P' et Q' sont faciles à déterminer, ce sont les points d'intersection de l'horizon fronto-parallèle (de rayon $\frac{|l+m|}{l}$) avec la droite de direction $\vec{T} = (T_x, T_y)$

dans le plan image :

$$\begin{aligned} x &= \pm \frac{l+m}{l} \frac{T_x}{\|\vec{T}\|} \\ y &= \mp \frac{l+m}{l} \frac{T_y}{\|\vec{T}\|} \end{aligned} \quad (3.36)$$

En se servant de la projection inverse sur la sphère définie dans le système 2.40, il est possible de calculer les coordonnées des points A , P et Q sur la sphère. Le

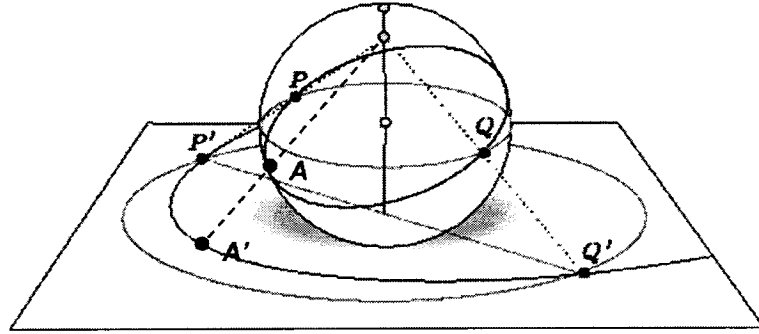


FIG. 3.10 Construction de la conique passant par un point de l'image par projection d'une droite dans la direction de translation

vecteur \vec{n} normal au plan formé par les points P , Q et A est défini par :

$$\vec{n} = \frac{\overrightarrow{AP} \times \overrightarrow{AQ}}{\|\overrightarrow{AP}\| \|\overrightarrow{AQ}\|} \quad (3.37)$$

Nous avons dès à présent les paramètres de l'ellipse passant par A' , P et Q , en se servant du système 2.48. Pour trouver le vecteur tangent \vec{t} , il est plus aisé de

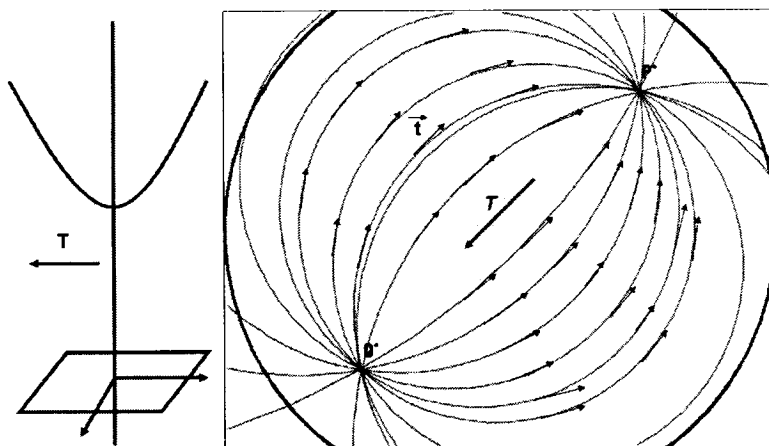


FIG. 3.11 Direction et sens du flux optique pour une translation T dans un plan perpendiculaire l'axe de la caméra.

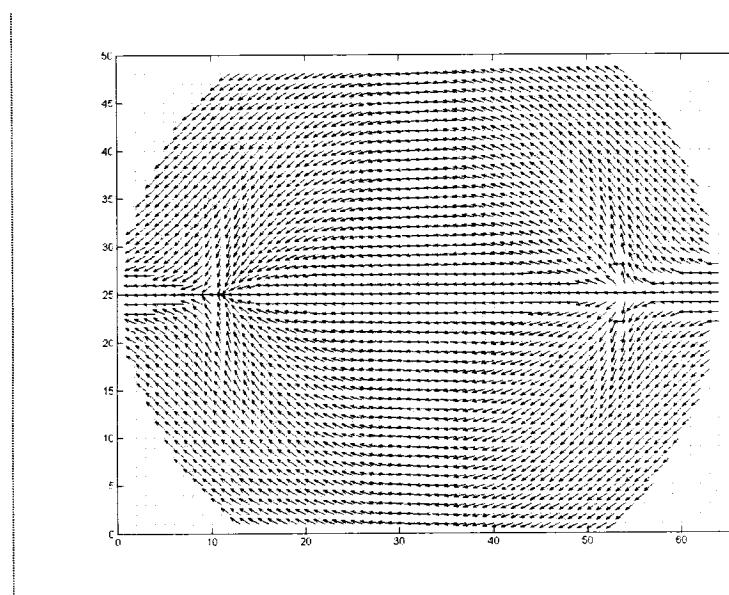


FIG. 3.12 Vecteurs flux théoriques dans le cas d'une translation perpendiculaire à l'axe de la caméra, un vecteur sur 10 est affiché.

connaître l'équation de l'ellipse dans le repère lié au plan image. Il faut utiliser le centre de l'ellipse $O' = [x_0, y_0]^T$, dont les coordonnées sont calculées au chapitre 2, ainsi que l'angle θ de la rotation du repère. Soit $A' = [x, y]^T$ dans le repère centré au centre de l'image omnidirectionnelle. L'équation de l'ellipse s'écrit alors dans le repère de l'image :

$$\frac{(\cos \theta(x - x_0) - \sin \theta(y - y_0))^2}{a^2} + \frac{(\sin \theta(x - x_0) + \cos \theta(y - y_0))^2}{b^2} = 1$$

Le vecteur tangent \vec{t} , illustré à la figure 3.11 s'écrit finalement :

$$\vec{t} = \begin{pmatrix} \frac{\sin \theta}{a^2}(\cos \theta(x - x_0) - \sin \theta(y - y_0)) - \frac{\cos \theta}{b^2}(\sin \theta(x - x_0) + \cos \theta(y - y_0)) \\ \frac{\cos \theta}{a^2}(\cos \theta(x - x_0) - \sin \theta(y - y_0)) + \frac{\sin \theta}{b^2}(\sin \theta(x - x_0) + \cos \theta(y - y_0)) \end{pmatrix} \quad (3.38)$$

Dans ce cas très particulier l'orientation théorique du flux peut-être illustré plus précisément (voir figure 3.12).

Si l'orientation pour chaque pixel est calculé, la moyenne sur toute l'image correspond à l'angle de la translation.

3.5 Système utilisé

Comme nous allons passer aux tests de validation sur des images il s'agit de présenter le système catadioptrique utilisé. Nous avons choisi d'utiliser un miroir hyperboloïdal car comme détaillé dans le chapitre 2, une caméra conventionnelle peut être utilisée, contrairement au miroir paraboloidal qui nécessite une lentille orthographique. Une caméra de type webcam a été choisie pour sa simplicité d'installation et d'utilisation (seulement un câble USB pour l'acquisition de données



FIG. 3.13 Miroir hyperboloïdal utilisé pendant les expérimentations.

TAB. 3.1 Caractéristique de la caméra omnidirectionnelle.

Caméra	QuickCam Pro 4000
Résolution	640x480
Distance Focale	807 pixels
Miroir	Hyperboloïdale
Fournisseur	NeoVision
Grand axe a	28.0950 mm
Petit axe b	23.4125 mm
Distance entre les deux foyers $2c$	73.1429 mm

et l'alimentation). Une photo du miroir utilisé est présentée figure 3.13 et le tableau 3.1 énumère les paramètres du miroir (obtenu auprès du fournisseur) et de la caméra (obtenu grâce au logiciel de calibration «Camera Calibration Toolbox» de Matlab).

3.6 Validité du calcul de flux optique

Nous avons décrit dans un premier temps la technique de calcul de flux choisie, puis notre technique pour calculer les différents opérateurs utiles pour ce calcul. Il s'agit maintenant de tester la validité de ce calcul. L'amplitude et l'orientation du flux optique doivent être en accord avec la structure de la scène et les différents mouvements, que ce soit de la caméra ou d'objets dans la scène. (Barron, 2005) suggère deux études de performance de calcul de flux.

La première est qualitative et permet un jugement général sur le flux représenté. Il s'agit d'étudier le flux quand aucune information sur la scène ou les mouvements sont connus mais cela ne permet pas de comparer l'algorithme utilisé avec d'autres. Elle permet une confirmation intuitive du calcul de flux.

La seconde est quantitative et doit s'appuyer sur des informations sur la scène et sur les mouvements. Cela peut-être des informations sur l'amplitude ou l'orientation du mouvement d'un objet dans la scène 3D, ou bien des informations sur le type de mouvement, l'amplitude et l'orientation du mouvement de la caméra. Pour réaliser cette étude, nous nous sommes appuyés sur trois types de donnée.

- Tout d'abord des images synthétiques simples, représentant des mouvements de formes dans le plan image. Ces images ont permis une étude en orientation et en amplitude du flux optique.
- Ensuite, un environnement 3D a été créé et a permis de simuler trois types de mouvement d'une caméra omnidirectionnelle : translation dans l'axe de la caméra, translation perpendiculaire à l'axe de la caméra et rotation par rapport autour de l'axe de la caméra. L'étude de l'amplitude du flux optique n'a pas été réalisée sur ces simulations et nous nous sommes contentés d'une étude en orientation du flux. Cet aspect de l'étude manque cruellement à notre étude et nous nous sommes efforcés de la compenser par une étude sur l'orientation du flux. L'étude de la validité du calcul de l'amplitude du flux optique nécessite

l'estimation de l'amplitude théorique du flux optique, valeur que nous n'avons obtenu que dans le cas des figures simples précédentes. Il aurait fallu obtenir la distance de tous les points objets de la scène 3D. C'est cette information qui nous a manquée pour estimer l'amplitude théorique du flux et le comparer à l'amplitude obtenu avec les algorithmes de calcul de flux.

- Finalement, des images réelles ont été réalisées grâce à la caméra omnidirectionnelle décrite précédemment pour les mêmes trois types de mouvement.

Si le choix des trois types de mouvements peut paraître restrictif, il n'en est rien. La plupart des études de flux avec des caméras omnidirectionnelles décrites dans la littérature ne considère que des translations perpendiculaires à l'axe de la caméra ou des rotations autour de l'axe de la caméra. De plus, ces études ne sont pas pixel par pixel et considèrent des caractéristiques générales comme l'orientation du foyer d'expansion ou du foyer de contraction. Nous proposons une étude, certes restrictive puisqu'elle ne s'appuie que sur l'orientation dans la majeure partie des images, mais en tout point de l'image, et avec un plus grand nombre de mouvements.

Nous utiliserons le terme de «OMNIFLUX» pour parler de l'algorithme écrit avec l'adaptation des opérateurs. A certains moments nous utiliserons un algorithme de calcul de flux optique utilisant la technique différentielle de Lukas et Kanade trouvé dans la librairie d'OPENCV. Les valeurs fournies par ce dernier permettent de donner une idée des ordres de grandeur attendus avec un algorithme de calcul de flux optique reconnu sur les images utilisées. Cet algorithme utilise les opérateurs suivants :

- pour la dérivation,

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (3.39)$$

– pour la matrice W de voisinage,

$$W = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (3.40)$$

3.6.1 Figures simples

L'étude de cas très simples comme celui d'un carré se déplaçant dans l'image permet souvent de commencer une étude. Cela permet d'évaluer en amplitude et en orientation le flux optique retourné par le programme. Malheureusement, un cas comme celui-là n'est pas tout à fait adapté à l'analyse de notre algorithme de part la présence des déformations créées par le miroir. En effet, les opérateurs s'adaptent en fonction de la position et leur distribution vont traiter des objets de l'espace projetés sur le plan image. Nous devons utiliser des formes, certes simples, mais qui reflètent les déformations du miroir.

Par exemple pour l'opérateur de voisinage, de part et d'autre de son centre, sa distribution s'adapte à la résolution variable de l'image. Dans la figure 3.14, le dessin de gauche montre un opérateur symétrique qui pourra très bien gérer le cas d'un carré constant qui se déplace dans une image en perspective, alors que le dessin de droite présente un exemple d'opérateur centré à droite d'une image omnidirectionnelle. Sa partie droite (hachurée), couvre une région de l'espace aussi grande que la partie gauche (en pointillés). C'est pour cela que la partie de droite couvre moins de pixels sur l'image. Si nous souhaitons faire bouger une forme simple sur le plan image pour tester l'algorithme de calcul de flux, elle doit elle aussi refléter les déformations. Nous pouvons penser que le carré utilisé si souvent pour des images en perspective est un carré se déplaçant sur un plan parallèle au plan image. Reprenons cette idée et faisons évoluer une forme simple sur un plan parallèle au

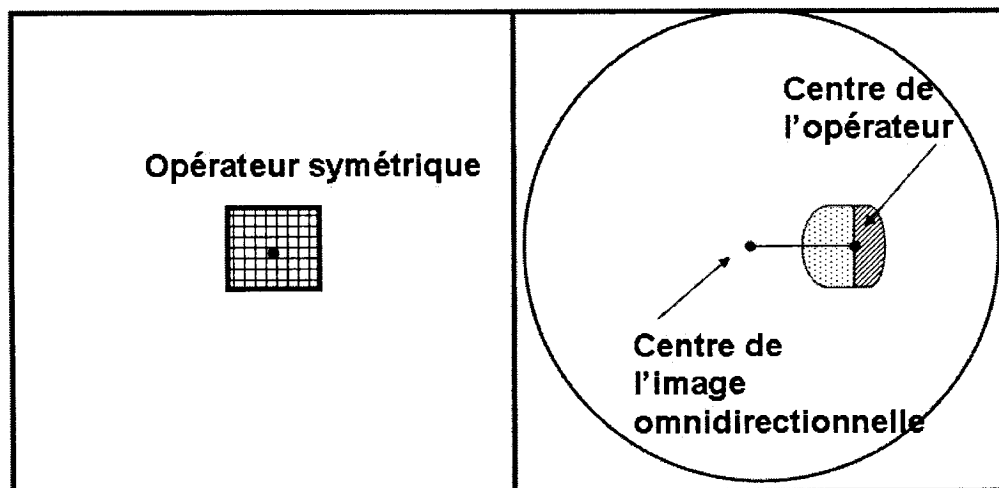


FIG. 3.14 Dessin de gauche : opérateur symétrique. Dessin de droite : opérateur de voisinage centré à droite de l'image omnidirectionnelle.

plan image. Nous connaissons son déplacement, son orientation, donc il sera facile d'évaluer l'amplitude et l'orientation du flux dans l'image.

Nous avons repris l'idée d'une gaussienne se déplaçant sur un plan $Z = Z_0$ et avons calculé le flux optique pour différentes positions dans l'image, différentes amplitudes de déplacement et différentes orientations.

3.6.1.1 Déplacement de la forme : premier exemple

La figure 3.15 fournit une illustration de la forme en déplacement, son centre se déplaçant du point de coordonnées $[0, 0]$ au point $[1, 1]$ (un déplacement d'un pixel de façon horizontale et un d'un pixel de façon verticale). Ceci n'est qu'une illustration et les résultats sont énumérés dans le tableau 3.2. Tout d'abord de façon qualitative, au niveau de la forme, les vecteurs du flux optique reflètent bien un

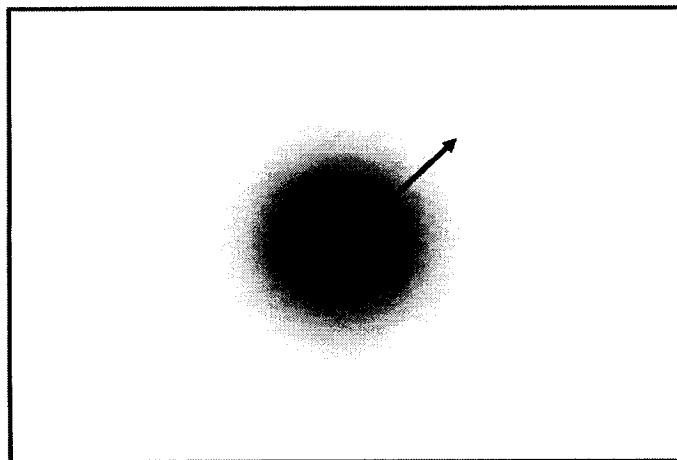


FIG. 3.15 Forme en déplacement.

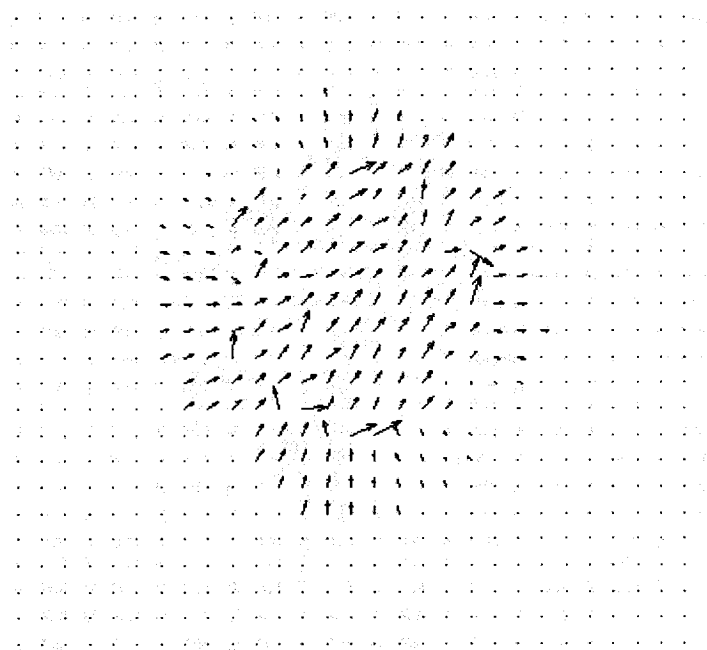


FIG. 3.16 Flux optique obtenu, un vecteur flux sur 10 est montré.

Résultats	OmniFlux	OPENCV
Amplitude moyenne	0.92 pixel	1.95 pixels
Écart type en amplitude	0.74 pixel	5.2 pixels
Erreur moyenne en orientation	13.5°	13.7°
Écart Type d'erreur en orientation	18.3°	17.66°

TAB. 3.2 Amplitude et erreur en orientation du champ de vecteurs de flux avec l'algorithme adapté et le programme de OPENCV.

déplacement de $[0, 0]$ vers $[1, 1]$. Les vecteurs de plus forte amplitude se situent à l'intérieur de la forme. D'autres vecteurs de très faible amplitude correspondent à la périphérie de la forme comme illustré sur la figure 3.16. Leur présence s'explique par l'utilisation d'opérateurs de grande taille (rappelons que notre taille maximale est 11 par 11) : même un point éloigné de quelques pixels de la forme sera couvert par l'opérateur de voisinage et ceux de dérivation et l'algorithme donnera un vecteur flux non nul en ce point.

Pour ce déplacement de $\sqrt{2}$ pixels en diagonale, les résultats sont présentés dans le tableau 3.2 : L'amplitude moyenne est la moyenne sur toutes les amplitudes de vecteurs de flux pour les pixels constituant la forme sur les deux images. Notre algorithme fournit une moyenne d'amplitude un peu faible mais proche des $\sqrt{2} \simeq 1.42$ pixels attendus. Les résultats en orientation sont bons et nous pouvons les illustrer sur la figure 3.17. Cette figure montre le champ obtenu après normalisation. Le champ théorique calculé grâce à la section 3.4.3 est illustré sur la figure 3.18.

3.6.1.2 Changement de position dans l'image et variation d'amplitude

Les opérateurs possèdent une symétrie radiale dans l'image omnidirectionnelle. Nous avons donc étudié seulement un rayon de l'image et nous avons choisi les positions $[0, 0]$, $[50, 50]$, et $[100, 100]$. Pour chaque position nous avons utilisés plusieurs amplitudes et les données sont rassemblées dans le tableau 3.3. Nous avons aussi choisi deux directions de déplacements pour le centre de la forme, un horizon-

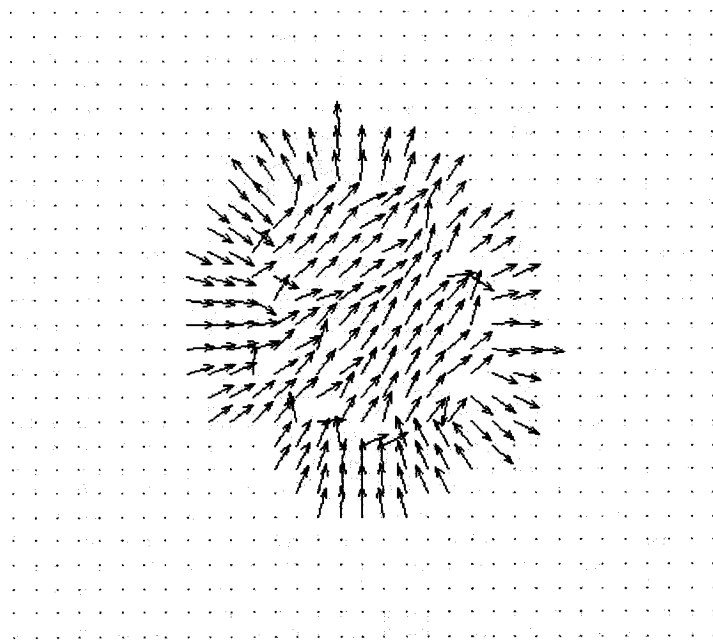


FIG. 3.17 Champ de vecteurs obtenus par adaptation des opérateurs avec normalisation.

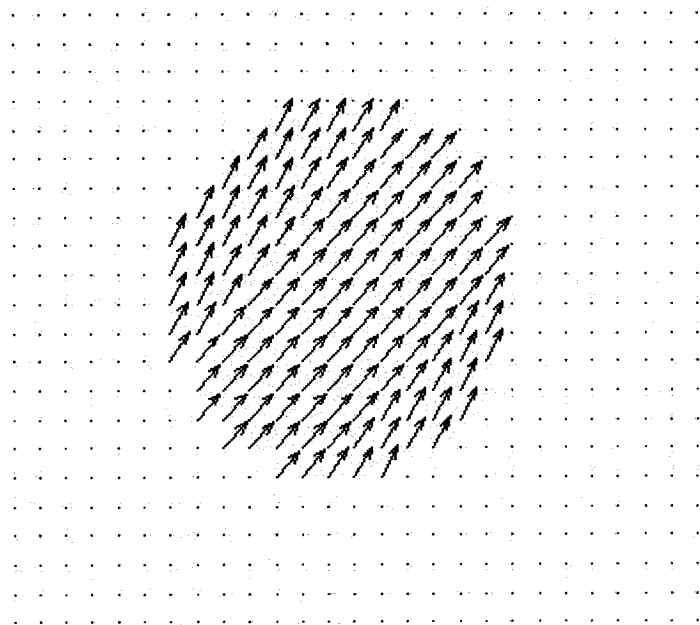


FIG. 3.18 Champ de vecteurs théoriques normalisés.

tal de 1, 2 ou 3 pixels, et un en diagonale avec n à la verticale et n à l'horizontale pour n égal à 1, 2 ou 3.

La première remarque à faire à la lecture de ce tableau est que les taux d'erreurs constatés avec notre algorithme sont du même ordre de grandeur que ceux d'un algorithme reconnu comme étant fiable, celui d'OPENCV.

La deuxième remarque est que notre programmation a permis d'éviter certains taux d'erreurs très élevés. Nous rappelons qu'à la fin du calcul de flux optique une inversion de matrice est présente et il faut veiller au conditionnement de la matrice pour ne pas voir le flux optique obtenir une amplitude exagérée (les techniques pour améliorer le conditionnement d'une matrice peuvent être trouvées dans (Dion, 1996)).

La troisième remarque est que l'erreur en amplitude grandit avec l'amplitude du déplacement. Cela se voit à toutes les positions dans l'image.

La quatrième remarque est que la qualité du calcul du flux diminue avec l'éloignement du centre de l'image. Néanmoins, cette diminution est moins flagrante que dans notre algorithme de référence. Plus la forme est loin du centre plus sa distribution va changer au cours du déplacement imposé. Pour l'illustrer, la forme ne change pas de façon significative au cours du déplacement dans le haut de la figure 3.19. Par contre, dans le bas de la figure 3.19, la forme change plus nettement. Au centre de l'image, les déformations dues au miroir ne sont pas significatives, alors qu'elles le deviennent en s'éloignant du centre. Notre algorithme s'adapte à ces changements de forme car la forme de ses opérateurs est conçu dans ce sens.

Enfin, les erreurs angulaires varient entre $12,5^\circ$ et 45° . Néanmoins, les plus grandes

Déplacement de la forme	Algorithme utilisé	ErAmp	EtAmp	ErO	EtO
[0,0] à [1,1]	OMNIFlux	0.75	0.47	13.4°	18.3°
	OPENCV	0.76	5.2	13.7°	17.6°
[0,0] à [2,2]	OMNIFlux	1.42	0.54	15.31°	19.2°
	OPENCV	2.07	14.05	18.8°	22°
[0,0] à [3,3]	OMNIFlux	1.26	3.42	18.3°	20.3°
	OPENCV	3.3	17.14	21.2°	21.72°
[0,0] à [1,0]	OMNIFlux	0.52	0.35	14°	19.7°
	OPENCV	0.45	2.9	12.5°	17.21°
[0,0] à [2,0]	OMNIFlux	1.04	0.61	14.1°	18.6°
	OPENCV	1.48	10.6	18.42°	22.5°
[0,0] à [3,0]	OMNIFlux	1.62	1.54	14.7°	17.4°
	OPENCV	2.5	15.9	21.2°	24.2°
[50,50] à [51,51]	OMNIFlux	0.58	0.47	17.3°	19.5°
	OPENCV	0.55	3.65	19.3°	20.5°
[50,50] à [52,52]	OMNIFlux	1.52	3.46	23.8°	23.1°
	OPENCV	1.9	19	24.7°	21.5°
[50,50] à [53,53]	OMNIFlux	2.6	7.2	27.4°	23.1°
	OPENCV	3.3	31.2	28.3°	21.8°
[50,50] à [51,50]	OMNIFlux	0.45	0.44	16°	19°
	OPENCV	0.28	1.26	15.1°	18.1°
[50,50] à [52,50]	OMNIFlux	0.86	0.87	16.4°	18.1°
	OPENCV	0.63	3.13	19.8°	21.7°
[50,50] à [53,50]	OMNIFlux	1.56	2.58	18.4°	17.7°
	OPENCV	1.6	14.3	22.7°	22.5°
[100,100] à [101,101]	OMNIFlux	3.4	16.1	39.8°	27.7°
	OPENCV	3.6	15.4	38°	26.5°
[100,100] à [102,102]	OMNIFlux	1.73	8.08	35.5°	25.4°
	OPENCV	8.06	31.6	43°	25.2°
[100,100] à [103,103]	OMNIFlux	2.8	12	39.4°	26.1°
	OPENCV	12.8	51.1	46.6°	24.4°
[100,100] à [101,100]	OMNIFlux	0.4	1.7	42°	26.8°
	OPENCV	1.72	7.2	34°	23.2°
[100,100] à [102,100]	OMNIFlux	0.9	3.6	42.5°	26.7°
	OPENCV	3.8	20	36.8°	24.2°
[100,100] à [103,100]	OMNIFlux	1.45	5.1	43.1°	27.5°
	OPENCV	5.5	24.8	38°	25.3°

TAB. 3.3 Amplitude et erreur en orientation du champ de vecteurs de flux avec l'algorithme adapté et le programme de OPENCV. ErAmp = Erreur moyenne en amplitude, EtAmp = Écart type de l'erreur en Amplitude, ErO = Erreur en Orientation, EtO = Écart type de l'erreur en orientation.

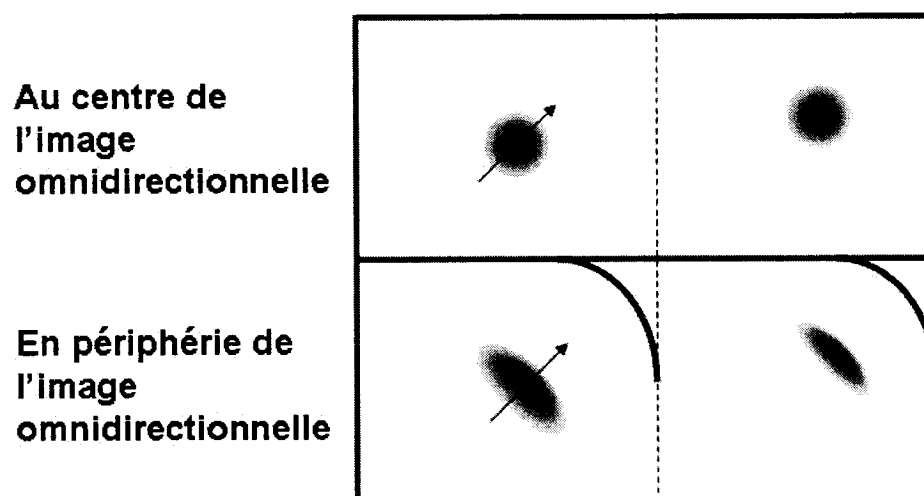


FIG. 3.19 Dessin du haut : le déplacement de la forme n'affecte pas sa forme au centre de l'image. Dessin du bas : le déplacement de la forme affecte la forme à la périphérie alors qu'il s'agit de la même amplitude dans l'espace et de la même direction.

erreurs surviennent pour les déplacements les plus grands et nous avons bon espoir d'avoir un taux d'erreur acceptable si nous restons sur des déplacements faibles.

3.6.2 Validation sur des images synthétiques

Pour continuer la validation, nous avons utilisé un environnement virtuel 3D. Une caméra virtuelle a été créée en respectant les mêmes valeurs de focale et de résolution que la caméra omnidirectionnelle du laboratoire. Les valeurs numériques seront trouvées dans le tableau 3.1.

Nous souhaitons rappeler au lecteur que les études de validité du calcul du flux qui vont suivre ne concernent que la comparaison de l'orientation du flux obtenu avec notre méthode avec l'orientation théorique du flux. Pour une étude de l'amplitude du flux, il aurait fallu connaître les distances de tous les points de l'espace 3D par rapport à la caméra, pour pouvoir évaluer l'amplitude théorique du flux optique. Les vecteurs de flux qui seront présentés seront normalisés pour une meilleure visualisation et permettront l'analyse du sens et de la direction du flux optique, c'est-à-dire de son orientation. Nous avons été incapables d'estimer l'amplitude théorique du flux dans une séquence d'images dans le cadre d'une caméra omnidirectionnelle et nous présentons donc l'analyse sur seulement une des composantes du flux, son orientation. Une étude future pourrait consister à générer de façon automatique des formes en stipulant les positions exactes de tous les points les constituant. Seulement alors nous pourrions estimer le flux optique théorique dans une séquence d'images filmant ces objets.

L'environnement est une salle cubique dans lequel sont disposés plusieurs objets et comme nous avons la possibilité d'affecter différentes textures aux différents objets et au pan de murs, nous avons opté pour deux configurations.

La première, appelée *configuration 1* est illustrée sur la figure 3.20 et présente pour les objets les caractéristiques suivantes :

- une sphère suspendue visible dans la partie gauche de l'image,
- un pavé sur le sol visible dans le coin droit de l'image dont les faces ne sont pas très texturées,
- un premier cube posé sur le pavé précédent dont les faces ne sont pas très texturées,
- un deuxième cube suspendu en haut à droite de l'image dont les faces ne sont pas très texturées,
- un troisième cube sur le sol près de la caméra au centre en bas de l'image et ses faces ne sont pas très texturées,
- et un cône posé sur le sol visible à gauche de l'image,
- deux cylindres disposés debout au fond de la salle virtuelle,
- les murs sont tous très texturés, de même que le plafond et le sol.

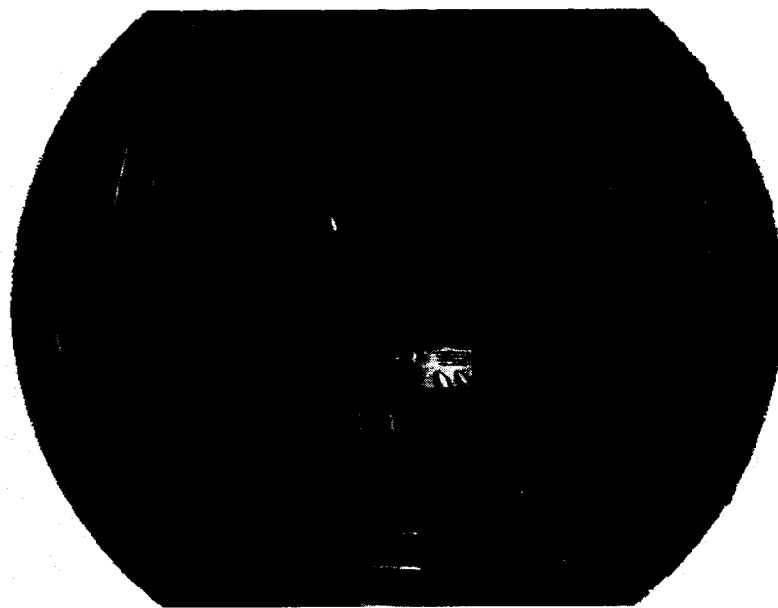


FIG. 3.20 Environnement créé pour la validation calcul de flux optique : *configuration 1*.

La *configuration 2*, présente à un cube près les mêmes objets mais pas texturés de la même façon (voir figure 3.21) :

- une sphère suspendue visible dans la partie gauche de l'image et est texturée,

- un pavé sur le sol visible dans le coin droit de l'image dont les faces sont très texturées,
- un premier cube posé sur le parallélépipède dont les faces sont très texturées,
- un troisième cube sur le sol près de la caméra au centre en bas de l'image dont les faces sont unies,
- un cône posé sur le sol visible à gauche de l'image,
- deux cylindres disposés debout au fond de la salle virtuelle,
- les murs sont tous très texturés, mais le plafond et le sol sont unis.

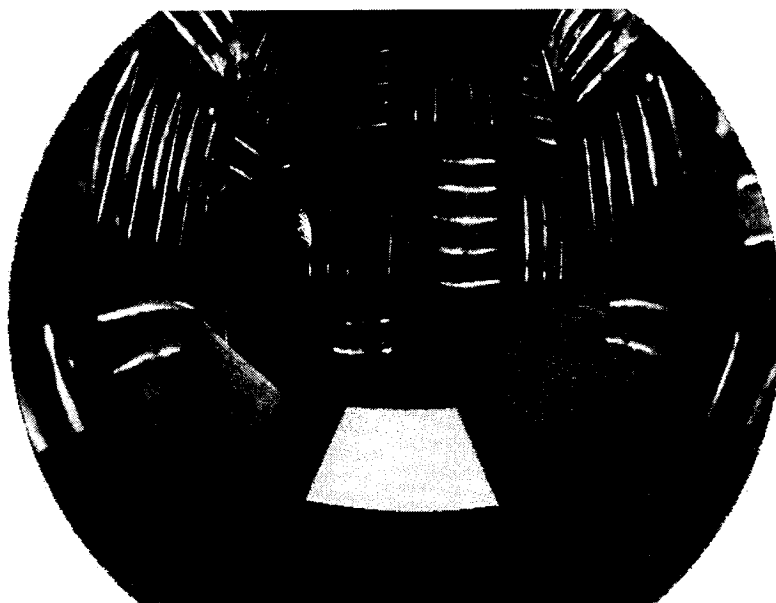


FIG. 3.21 Environnement créé pour la validation calcul de flux optique : *configuration 2*.

Une autre problème est de choisir des amplitudes pour les mouvements. Non seulement l'amplitude du flux optique va dépendre de la position des objets dans l'espace, mais les déformations du miroir augmentent ce phénomène. Comme nous avons décidé de faire une étude sur l'orientation du flux optique, il fallait choisir des amplitudes de déplacements qui génèrent dans l'image un flux optique non nul quel que soit la position dans l'image. L'amplitude du flux va dépendre de la distance à la caméra des objets observés. Nous parlons ici d'images où les objets sont

un minimum texturés car une zone parfaitement unie fournira un flux nul quelque soit distance. Après plusieurs essais, nous avons opté pour les séquences suivantes, qui comme nous le verrons plus tard fournissent des flux non nuls dans toutes les parties de l'image :

- translation dans l’axe de la caméra de 1cm,
- rotation d’un degré,
- translation de 1cm perpendiculairement à la caméra (de façon horizontale par rapport à l’axe de la caméra).

Les champs de vecteurs calculés avec notre algorithme sont dessinés pour les trois types de mouvements dans les figures 3.22, 3.23 et 3.24 pour la *configuration 1* et dans les figures 3.25, 3.26 et 3.27 pour la *configuration 2*. La figure 3.24 affiche aussi certaines des ellipses sur lesquelles devraient reposer les vecteurs de flux optique. La partie de l’ellipse à considérer est celle située entre les points P et Q . De plus le point P est ici le foyer de contraction et le point Q représente le foyer d’expansion. Pour une meilleure visualisation et puisque c’est l’orientation que nous allons étudier en détail, nous avons présenté des vecteurs normalisés. Les points où les vecteurs sont nuls correspondent donc à des pixels d’amplitude très faibles.

Il arrive que dans certaines zones non texturées des images synthétiques, l’amplitude du flux calculé soit nulle. La phase est alors indéterminée. Nous avons choisi d’afficher seulement un point sur les figures précédentes pour les points en question. Ces schémas sont là pour illustrer la direction et le sens des vecteurs de flux optique.

Pour la *configuration 1*, les objets dont les faces étaient unies présentent des vecteurs de flux nuls. C’est le cas pour le cube posé au sol juste devant la caméra ou pour le cube suspendu, pour le cube posé sur le parallélépipède et pour le parallélépipède. Dans les figures 3.23 et 3.24, une partie du cône et de la sphère suspendue présentent des vecteurs de flux négligeables.

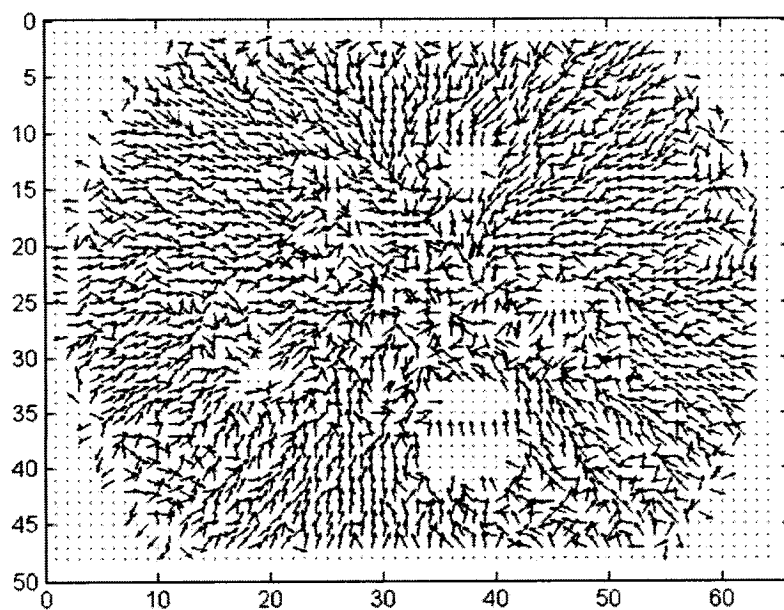


FIG. 3.22 *Configuration 1* : Représentation du flux optique pour une translation dans l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

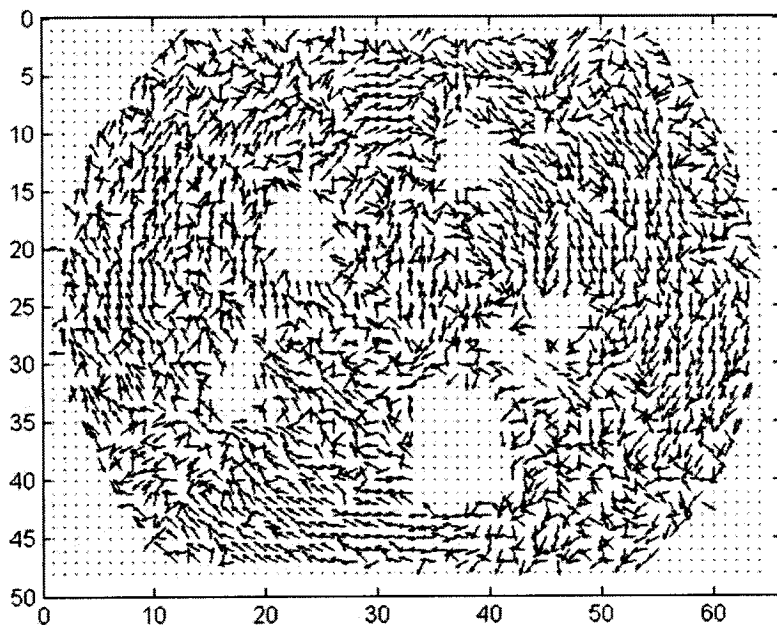


FIG. 3.23 *Configuration 1* : Représentation du flux optique pour une rotation autour de l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

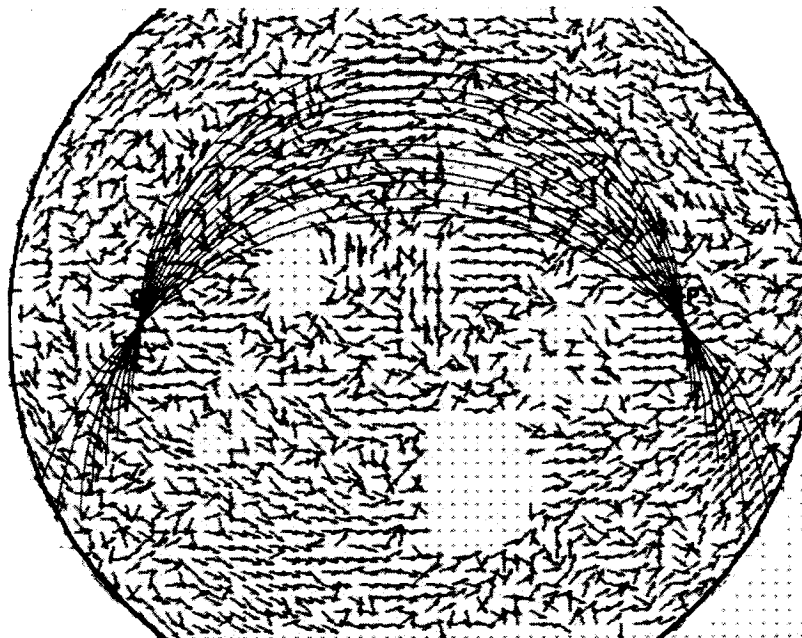


FIG. 3.24 *Configuration 1* : Représentation du flux optique pour une translation perpendiculaire à l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

Pour la *configuration 2*, le parallélépipède et le cube qui le surmonte sont maintenant très texturés et présentent des vecteurs de flux non nuls. Par contre, le plafond et le sol sont maintenant unis et il est difficile de calculer un flux optique significatif sur ces emplacements. Le cube placé devant la caméra a toujours des faces unies et le flux optique n'est important que sur ses arêtes comme le montre bien la figure 3.26.

Pour juger de la qualité de notre algorithme sur ces images, trois calculs ont été effectués sur des séquences d'images comprenant chacune deux images.

Tout d'abord l'orientation moyenne dans l'image est calculée, c'est-à-dire qu'en chaque point l'orientation est calculée et qu'elle est moyennée sur toute l'image. Dans le cas d'une translation perpendiculaire à la caméra, ce calcul permet d'estimer le foyer d'expansion dans l'image. Ce test est inspiré de (Daniilidis, 2002),

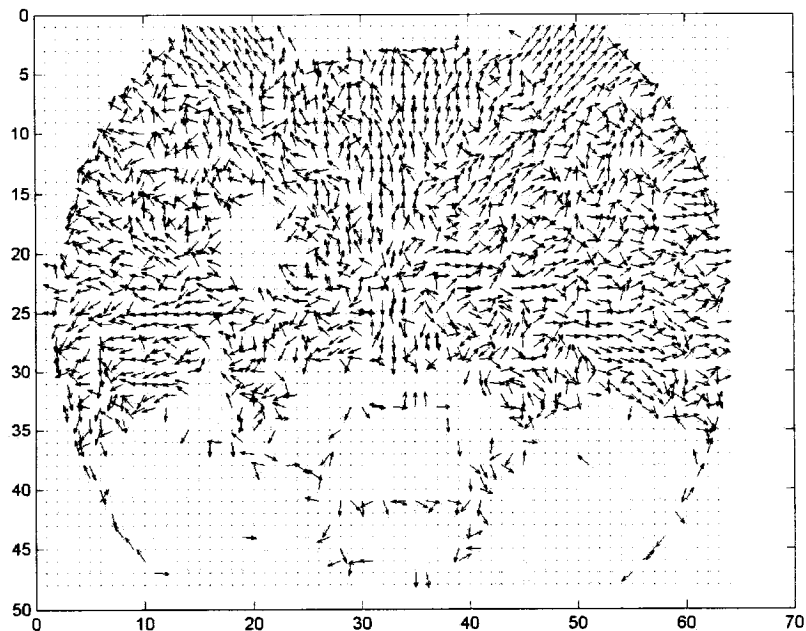


FIG. 3.25 *Configuration 2* : Représentation du flux optique pour une translation dans l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

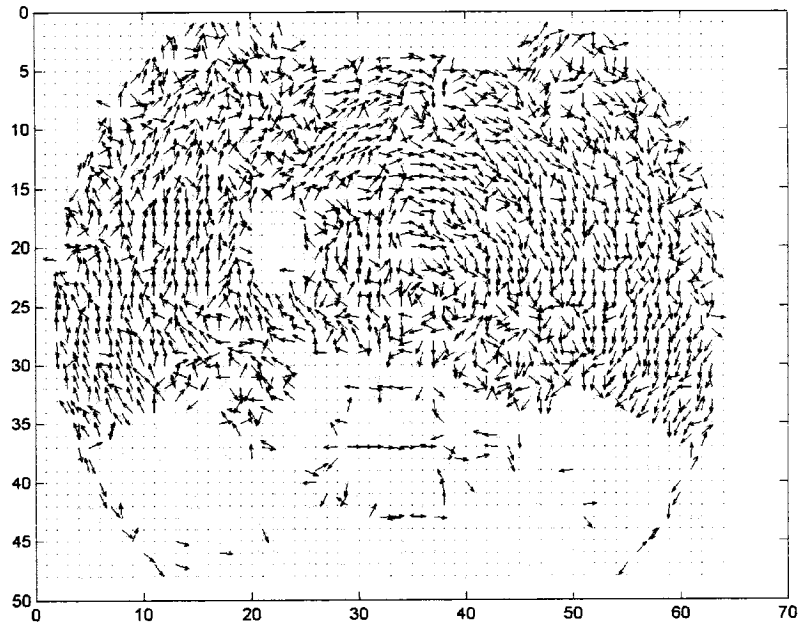


FIG. 3.26 *Configuration 2* : Représentation du flux optique pour une rotation autour de l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

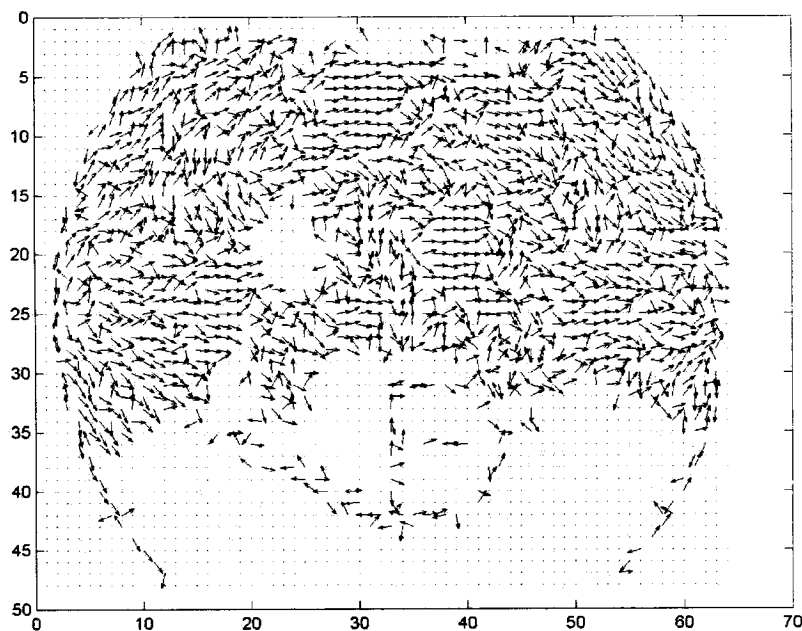


FIG. 3.27 *Configuration 2* : Représentation du flux optique pour une translation perpendiculaire à l'axe de la caméra. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

qui ne se sert que de cette valeur pour tester son calcul de flux dans une image omnidirectionnelle. Pour nos trois mouvements, la moyenne doit être nulle. En effet, pour la translation dans l'axe et la rotation, tous les vecteurs se compensent en orientation, et pour la translation perpendiculaire à l'axe, nous l'avons prise horizontale donc la moyenne est là aussi nulle.

Ensuite, le calcul de l'erreur en orientation (différence entre l'orientation obtenue par notre algorithme et l'orientation théorique calculée à la section 3.4) est calculée pour chaque pixel pour lequel l'amplitude du flux est non nulle. En effet, notre orientation théorique estimée dans la section 3.4, donne l'orientation théorique du flux lorsque celui-ci est non nul. Si l'amplitude du vecteur flux optique est nul, il est normal que sa phase soit indéterminée et ne soit pas prise en compte dans le calcul d'erreur. La moyenne sur tous les pixels en question est ensuite calculée. Enfin l'écart type de l'erreur en orientation est aussi calculé.

	configuration 1		configuration 2	
	OMNIFlux	OPENCV	OMNIFlux	OPENCV
Orientation moyenne	3.3°	-1.3°	4.6°	-2.8°
Erreur Moyenne en orientation	33.7°	28.2°	35.2°	42.5°
Écart Type	25.8°	28.2°	26.1°	28.5°

TAB. 3.4 Etudes des champs de flux optique obtenus pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV, pour une translation perpendiculaire à l'axe (de façon horizontale sur l'image considérée).

	configuration 1		configuration 2	
	OMNIFlux	OPENCV	OMNIFlux	OPENCV
Orientation moyenne	-4°	-4.6°	0.2°	-3.7
Erreur Moyenne en orientation	34.1°	39.6°	35.3°	36.9°
Écart Type	26.9°	29.4°	27.4°	29.2°

TAB. 3.5 Etudes des champs de flux optique obtenus pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV, pour une rotation autour de l'axe de la caméra.

Pour les trois mouvements et pour les deux configurations décrites précédemment, les calculs ont été effectués et sont rassemblés dans les tableaux 3.4, 3.5 et 3.6. De manière générale, les résultats obtenus avec notre algorithme sont cohérents avec ceux obtenus par l'algorithme d'OPENCV. Pour l'orientation, l'erreur en orientation pixel par pixel et moyennée et l'écart de cette erreur, les ordres de grandeur sont similaires et confirme l'analyse qualitative.

	configuration 1		configuration 2	
	OMNIFlux	OPENCV	OMNIFlux	OPENCV
Orientation moyenne	-8.1°	-6.3°	-20.3°	-14.1°
Erreur Moyenne en orientation	32.4°	37.1°	34.1°	41.28°
Écart Type	26.3°	29°	26.7°	29.3°

TAB. 3.6 Etudes des champs de flux optique obtenus pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV, pour une translation dans l'axe de la caméra.

Orientation moyenne sur toute l'image :

1. Pour la translation perpendiculaire à l'axe, les résultats sont bons puisque l'orientation est proche de 0° ,
2. Pour la rotation, nous pouvons dire que les résultats sont bons puisque l'orientation est proche de 0° .
3. Pour la translation dans l'axe de la caméra, l'orientation est faible (8.1°), dans le cas d'un environnement structurée (*configuration 1*) mais trop importante dans le cas de la *configuration 2*. Cela s'explique par le nombre trop important de zones unies où le flux est faible (voir figure 3.25). Les vecteurs ne se compensent plus et la moyenne en orientation diffère de 0° .

Erreur en orientation et écart type :

1. Pour la translation perpendiculaire à la caméra, l'erreur calculée est assez élevée, mais reste moins élevée que celle obtenue avec OPENCV. Puisqu'elle considère les vecteurs d'amplitude non nulle, il est normal qu'elle soit du même ordre de grandeur pour les deux configurations. De plus, si nous nous référons à des articles scientifiques qui comparent les performances de flux optique ((Barron, 1994) et (McCane, 2001)), une erreur de l'ordre de 30° obtenue sur tous les pixels n'est pas surprenante quand tous les pixels de l'image sont considérés.
2. Pour la translation dans l'axe de la caméra, les résultats sont du même ordre de grandeur que pour le mouvement précédent.
3. Pour la rotation autour de l'axe de la caméra, les résultats sont moins bons que ceux d'OPENCV mais du même ordre de grandeur.

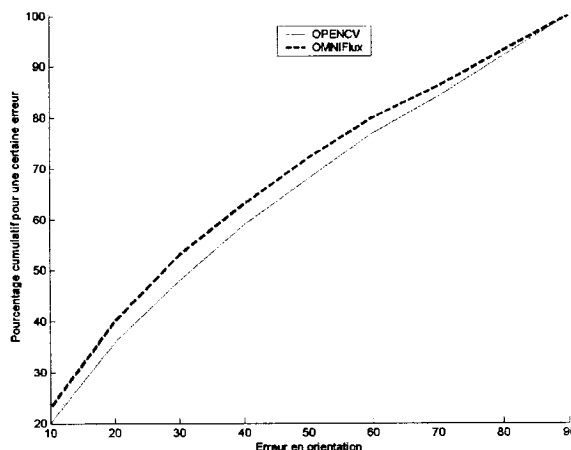


FIG. 3.28 Diagramme d'erreur cumulative pour une translation dans l'axe de la caméra.

L'algorithme développé pendant les recherches fournit des résultats du même ordre de grandeur qu'un algorithme reconnu, mais les taux d'erreurs restent quand même haut. Une autre façon de valider le calcul de flux est d'utiliser les diagrammes d'erreur cumulative décrits dans la section 3.3.2. Ils permettent au lecteur de se rendre compte que même si les taux d'erreurs moyens pour l'orientation sont élevés, un pourcentage non négligeable de l'image présente une erreur d'orientation assez faible. Pour le cas des trois mouvements décrits précédemment dans la *configuration 1*, les diagrammes d'erreur cumulative sont présentés sur les figures 3.28, 3.29 et 3.30. Pour les deux translations, la courbe de notre algorithme est placée en amont, c'est-à-dire qu'elle fournit pour le même taux d'erreurs, un pourcentage de pixels, qui présentent une erreur inférieure à ce taux, plus élevé. En d'autres termes l'algorithme est meilleur.

Dans les calculs précédents, la plupart des erreurs moyennes en orientation sont de l'ordre de 30° . Or pour chacun des diagrammes, les pixels avec une erreur en orientation inférieure à 30° représentent :

- pour la translation perpendiculaire à l'axe, 52% de l'image,

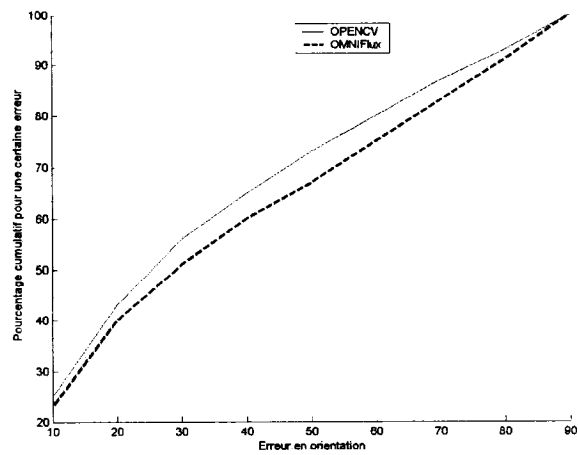


FIG. 3.29 Diagramme d'erreur cumulative pour une rotation dans l'axe de la caméra.

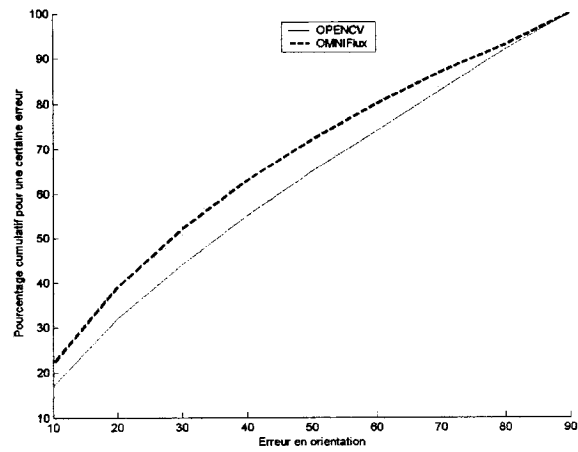


FIG. 3.30 Diagramme d'erreur cumulative pour une translation perpendiculaire à l'axe de la caméra.

	OMNIFlux	OPENCV
Orientation moyenne	-11.7°	-2°
Erreur Moyenne en orientation	45.6°	49°
Écart Type	29°	29°

TAB. 3.7 Translation dans l'axe de la caméra : études des champs de flux optique obtenues pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV.

	OMNIFlux	OPENCV
Orientation moyenne	-15.4°	-0.6°
Erreur Moyenne en orientation	44.5°	50.2°
Écart Type	28.2°	28.5°

TAB. 3.8 Translation perpendiculaire à l'axe : études des champs de flux optique obtenues pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV (translation horizontale sur l'image considérée).

- pour la translation dans l'axe, 53% de l'image,
- pour la rotation autour de l'axe de la caméra, 51% des pixels.

3.6.3 Validation sur des images réelles

La section précédente a validé le calcul de flux pour des images synthétiques. Nous avons aussi testé notre algorithme sur trois séquences de 2 images réelles correspondant aux trois mouvements suivants :

- une translation dans l'axe de la caméra de 2 cm,
- une translation perpendiculaire à l'axe de la caméra de 2 cm.
- une rotation de 2° autour de l'axe de la caméra.

Ces images ont été prises dans le laboratoire permettant le déplacement calibré de la caméra. Le laboratoire possède des locaux alternant zones très structurés (voir les étagères de la figure 3.31) et zones unies (le sol dans la figure 3.32). Les résultats sont présentés dans les tableaux 3.7, 3.8 et 3.9.

	OMNIFlux	OPENCV
Orientation moyenne	-8.1°	-0.4°
Erreur Moyenne en orientation	41.8°	46°
Écart Type	29°	29.2°

TAB. 3.9 Rotation autour de l'axe : études des champs de flux optique obtenues pour les deux configurations avec l'algorithme avec opérateurs adaptés, et celui d'OPENCV.

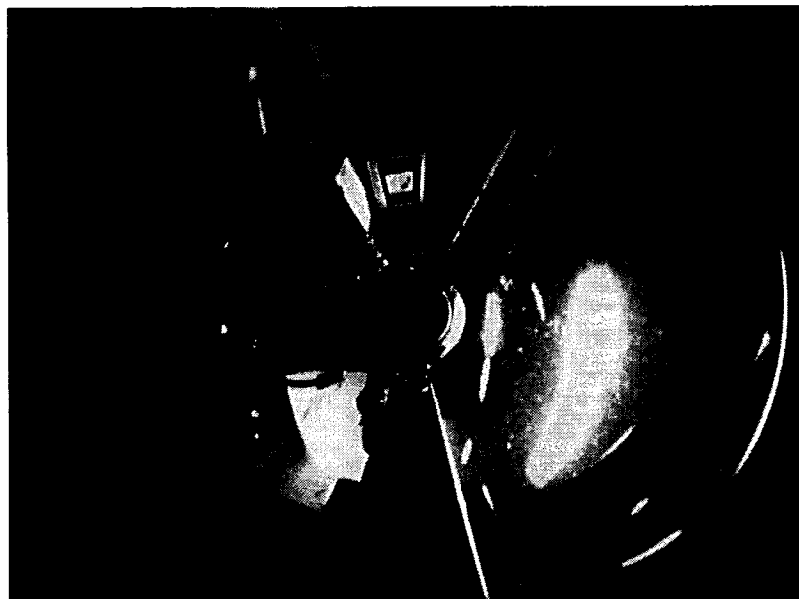


FIG. 3.31 Image extraite d'une séquence de translation perpendiculaire à l'axe de la caméra.

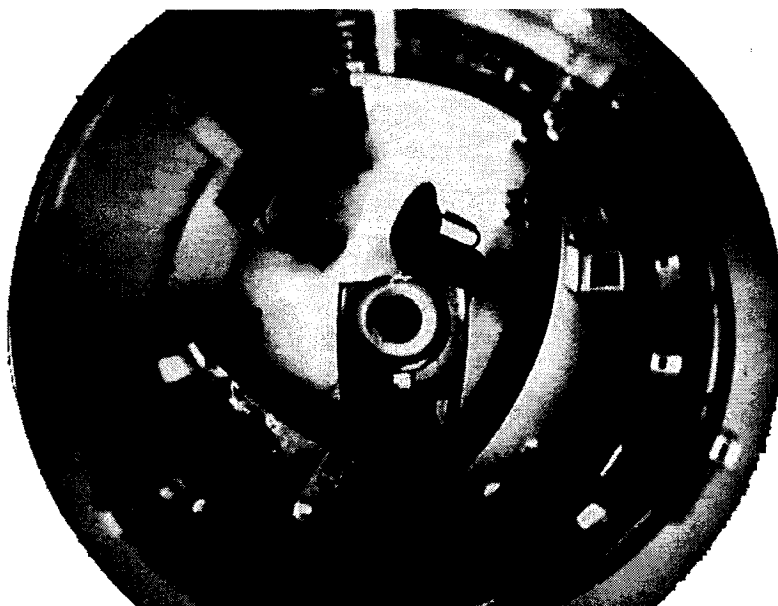


FIG. 3.32 Image extraite d'une séquence de translation dans l'axe de la caméra.



FIG. 3.33 Image extraite d'une séquence de rotation autour de l'axe de la caméra.

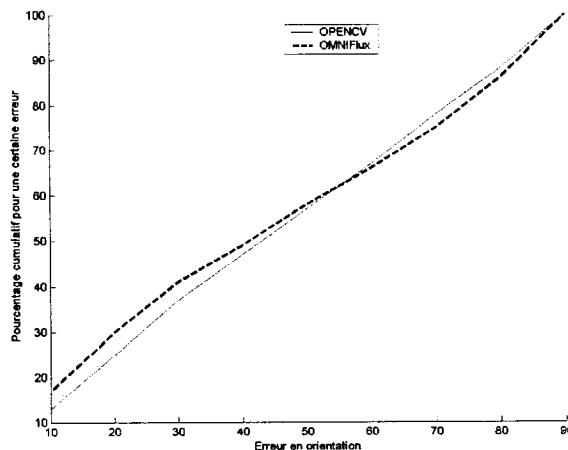


FIG. 3.34 Translation dans l'axe de la caméra : diagramme d'erreur cumulative.

La première remarque concernant les trois séquences est que les résultats sont moins bons avec des séquences réelles. Même si les écarts types sur les erreurs sont du même ordre qu'avec les images synthétiques, les erreurs moyennes sont légèrement supérieures.

Par contre, nous pouvons constater que les résultats sont toujours légèrement meilleurs qu'avec l'algorithme de référence.

La différence avec les images synthétiques réside dans les orientations moyennes. Elles sont dans le cas des images réelles supérieures à 10° alors qu'elles devraient être nulles.

Comme pour les images synthétiques, nous pouvons avoir recours aux diagrammes d'erreur cumulatifs pour valider notre algorithme. Les figures 3.34, 3.35 et 3.36 montrent les diagrammes pour les trois mouvements décrits. La validité de notre algorithme apparaît sur ces diagrammes de façon moins significative que pour les images synthétiques mais néanmoins réelle. Par exemple, pour une erreur en orientation de moins de 30° nous avons :

- pour la translation perpendiculaire à l'axe, 37% des pixels,

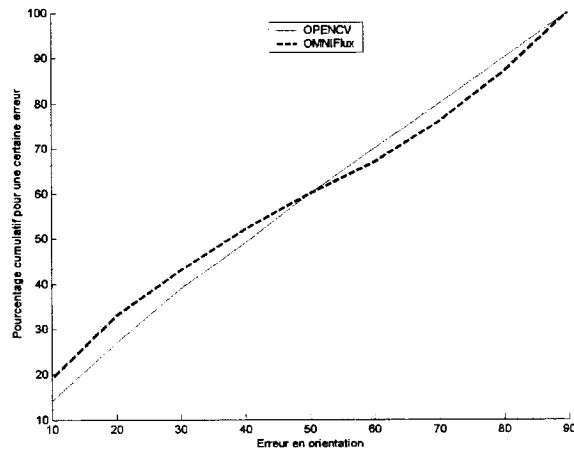


FIG. 3.35 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative.

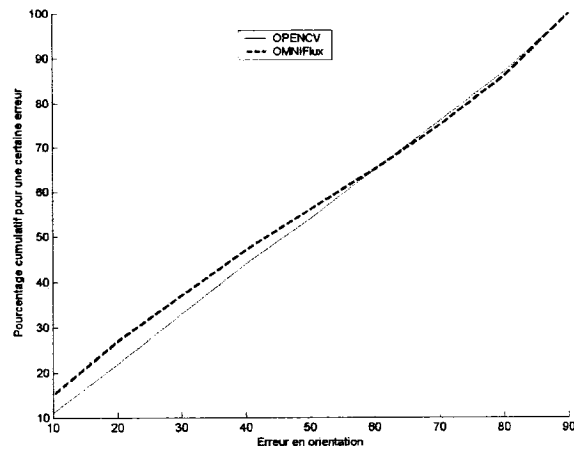


FIG. 3.36 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative.

- pour une translation dans l’axe, 41% des pixels,
- pour une rotation autour de l’axe de la caméra, 43% des pixels.

Nous venons d’étudier la validité de notre calcul de flux optique en trois points. Tout d’abord grâce à des formes simples se déplaçant sur le plan image. Elles ont permis une première étude en amplitude et en orientation. La suite de l’étude sur des séquences d’images synthétiques et réelles a permis d’aborder la validité de notre algorithme en ce qui concerne l’orientation. L’étude de l’amplitude des vecteurs de flux calculés reste un axe essentiel de l’étude du flux optique et qui n’a malheureusement pas été abordée.

Nous allons dans la prochaine section faire une étude comparative entre notre algorithme qui adapte ses opérateurs et les algorithmes utilisant des opérateurs dits homogènes, c’est-à-dire identiques en tout point de l’image. Cette étude reposera elle-aussi uniquement sur l’orientation du flux optique.

3.7 Comparaison avec les opérateurs homogènes

Cette section présente une comparaison de notre algorithme avec les différents algorithmes décrits dans la section 3.3.2, et qui utilisent des opérateurs homogènes dans toute l’image. Pour cela, les diagrammes d’erreur cumulative ont été tracés pour les trois cas de mouvements énoncés précédemment.

Cas des images synthétiques : Les calculs ont été effectués pour les séquences d’images synthétiques de la *configuration 1*. Toutes les séquences comportent seulement deux images par mouvement de caméra. Les mouvements sont identiques à ceux décrits dans la section 3.6.2. Les figures 3.37, 3.38 et 3.39. présentent les diagrammes d’erreur cumulative et conduisent à formuler les remarques suivantes.

- Dans les diagrammes 3.37 et 3.38 la courbe du pourcentage cumulatif de l’algo-

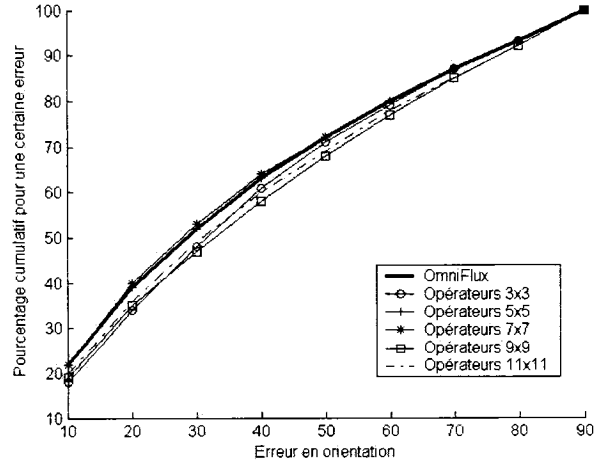


FIG. 3.37 Diagramme d'erreur cumulative pour une translation perpendiculaire à l'axe de la caméra.

rithme adapté est placée au dessus de toutes les autres. Il est donc le meilleur algorithme dans ces deux cas.

- Dans le cas de la rotation, notre algorithme est équivalent aux autres, sauf pour l'algorithme utilisant des opérateurs 11 par 11.
- Pour observer les diagrammes d'erreur cumulative, rappelons qu'il donner une attention particulière à la zone des taux d'erreur faible. C'est la zone où nous saurons le pourcentage de points de l'image correspondants à des taux vraiment faible, donc à un calcul vraiment juste de l'orientation. Dans le cas des diagrammes 3.37, 3.38 et 3.39, notre algorithme se confond avec d'autres, mais se situe soit en première position soit en seconde.

Cas d'images réelles : Les images et les mouvements sont identiques à ceux décrits dans la section 3.6.3 et toutes les séquences comportent deux images. Les études avec les séquences d'images réelles présentées précédemment ont montré que cet écart se réduit sur les images réelles. Les séquences d'images réelles présentent des similarités encore plus nettes entre les algorithmes. Les figures 3.40, 3.41 et 3.42 présentent les diagrammes d'erreur cumulative et conduisent à formuler les

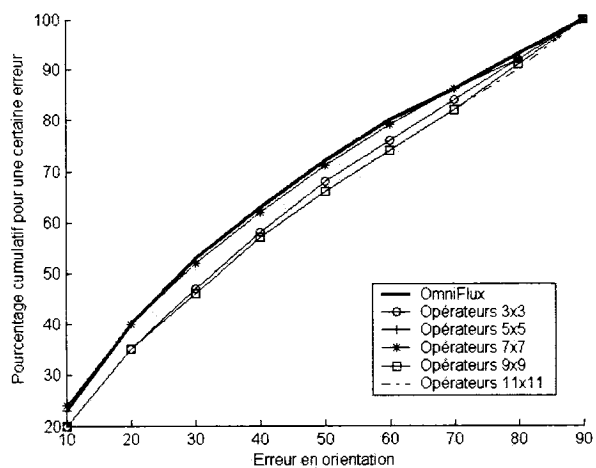


FIG. 3.38 Diagramme d'erreur cumulative pour une translation dans l'axe de la caméra.

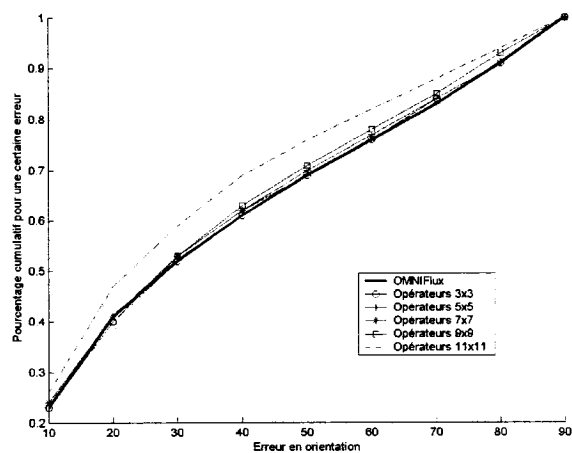


FIG. 3.39 Diagramme d'erreur cumulative pour une rotation autour de l'axe de la caméra.

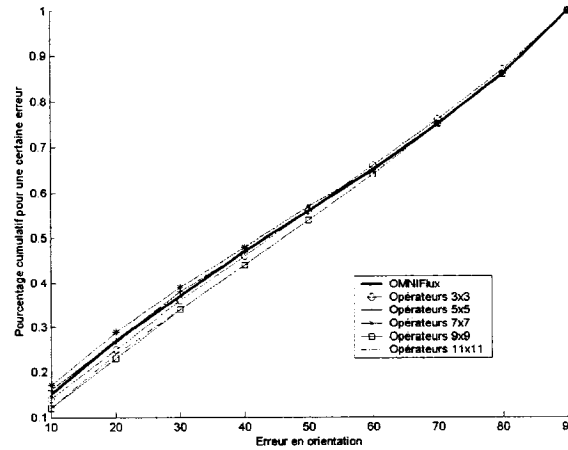


FIG. 3.40 Translation perpendiculairement à l'axe de la caméra : diagramme d'erreur cumulative.

remarques suivantes.

- L'écart entre les taux d'erreurs est faible.
- Néanmoins, en ce qui concerne les taux faibles, notre algorithme est bien placé même si ce n'est pas le meilleur. Il se place en seconde position dans les figures 3.40 et 3.41 et en troisième position dans la figure 3.42. Ces remarques sont valables pour des taux faibles mais plus les taux sont élevés, plus les courbes se rapprochent.
- En observant les diagrammes, il est clair que c'est l'algorithme 7 par 7 qui cumule les meilleurs courbes. Le bon sens permet d'imaginer qu'il est le juste milieu en tous les opérateurs qui vont de 3 par 3 à 11 par 11, et qu'il est normal qu'il présente en moyenne les meilleurs résultats.

Comme nos opérateurs évoluent en fonction de leur position dans l'image, une étude sur toute l'image ne fournit pas une preuve suffisante pour affirmer la prédominance d'un algorithme sur un autres. Nous avons vu qu'à partir d'un certain rayon dans l'image, les opérateurs devenaient identiques à ceux du groupe d'opérateurs 3 par 3 décrits dans la section 3.3.2. Si les effets de nos opérateurs sont importants avant

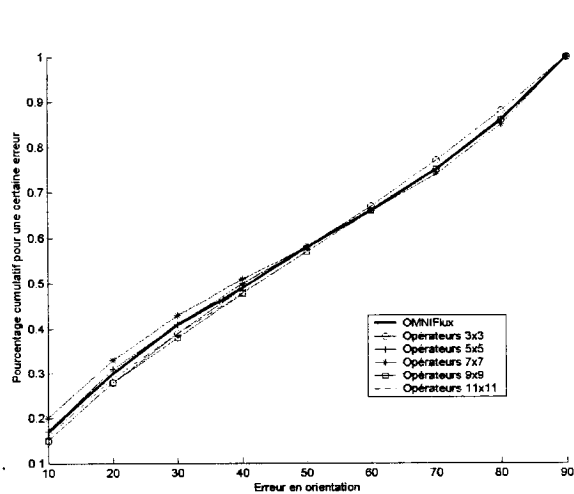


FIG. 3.41 Translation dans l'axe de la caméra : diagramme d'erreur cumulative.

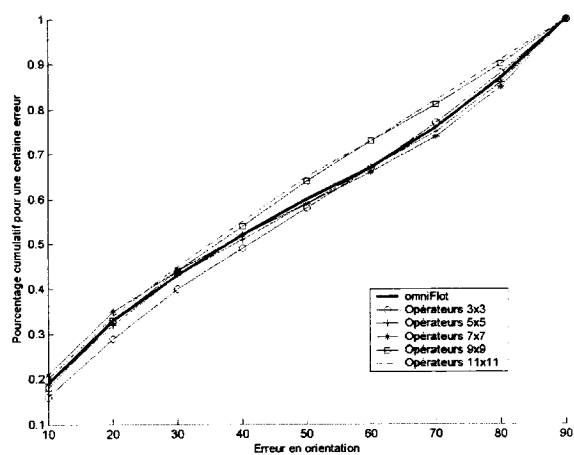


FIG. 3.42 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative.

ce rayon «limite», et si la moyenne est effectuée sur toute l'image, ses effets sont amoindris et il est difficile de les mettre en valeur. Il est donc indispensable d'étudier les diagrammes cumulatifs en fonction de la position dans l'image.

3.8 Observation par couronne

Il a fallu aussi effectuer des calculs d'erreurs en fonction de la position dans l'image et plus précisément en fonction du rayon. Ainsi des calculs d'erreur moyenne commise en orientation ont été effectués, non pas dans une image entière mais dans une couronne comprise entre deux rayons, R_1 et R_2 .

3.8.1 Séquences d'images synthétiques

Les figures 3.43 à 3.47 présentent l'erreur cumulative dans le cas de la translation dans l'axe de la caméra pour la *configuration 1* et pour une couronne de pixels donnée. Dans ces figures, il est possible d'établir les remarques suivantes :

- Pour une couronne entre 30 et 50 pixels (voir figure 3.43), notre algorithme présente pour des taux d'erreur faibles une bonne position.
- Dans les figures 3.44, 3.45 et 3.46, nous pouvons voir que l'algorithme adapté présente les meilleurs résultats. En particulier, entre 130 et 150 pixels, figure 3.45, il apporte une forte amélioration.
- Pour une couronne de 290 à 310 pixels (voir figure 3.47), l'algorithme adapté rejoint les opérateurs de petites tailles.

Si nous observons l'évolution des performances pour ce mouvement, nous pouvons dire que soit l'algorithme est à la première place, soit il est très bien placé. Néanmoins, il s'agit de rappeler que les différences de performance entre les algorithmes sont faibles.

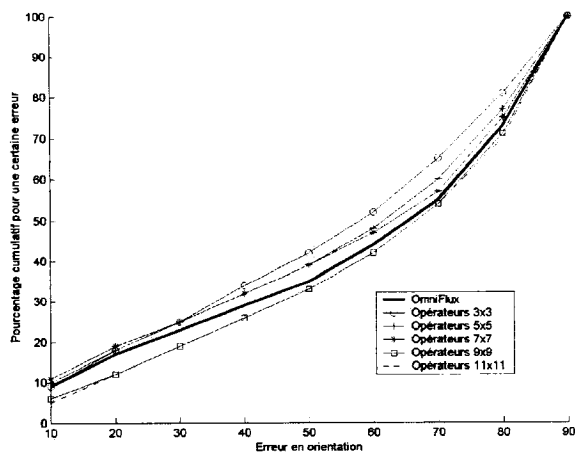


FIG. 3.43 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 30 et 50 pixels.

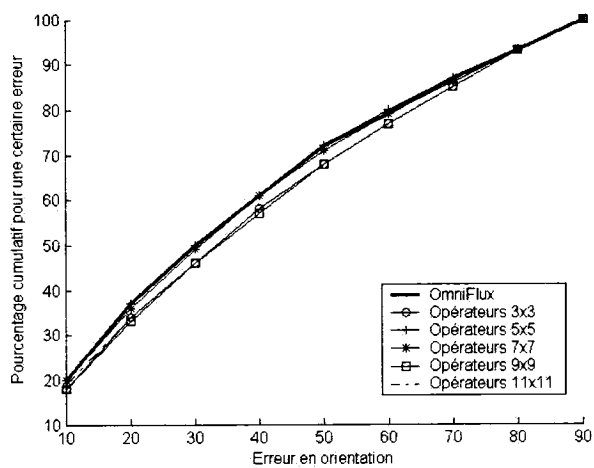


FIG. 3.44 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels.

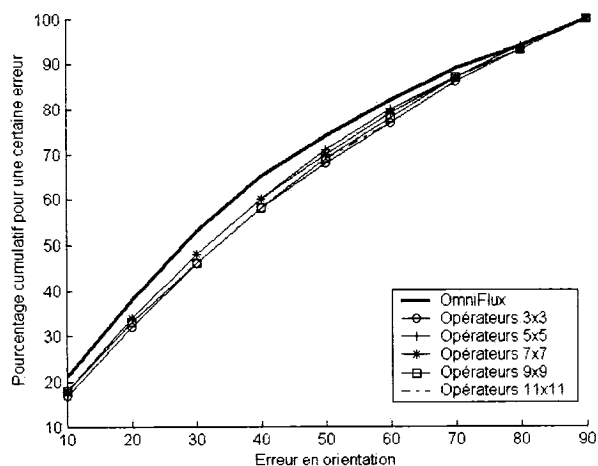


FIG. 3.45 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels

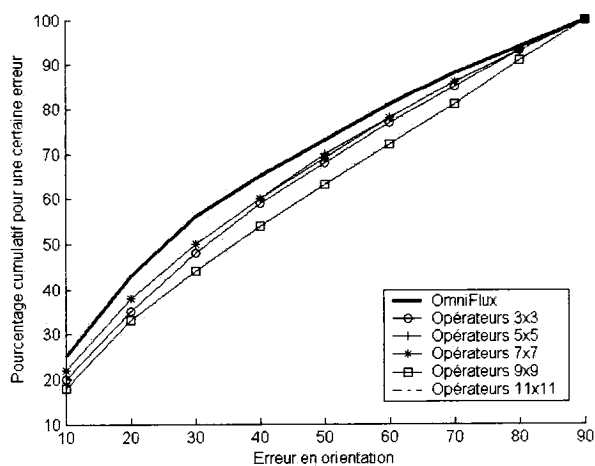


FIG. 3.46 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 150 et 170 pixels

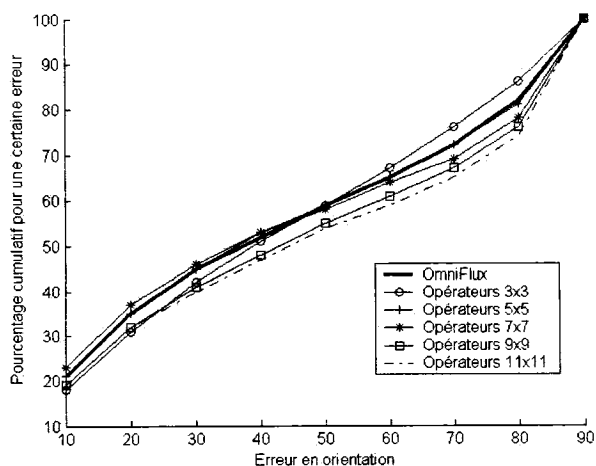


FIG. 3.47 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 290 et 310 pixels

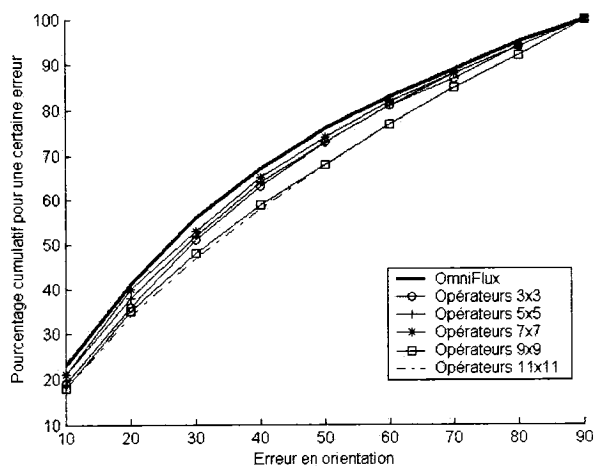


FIG. 3.48 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels.

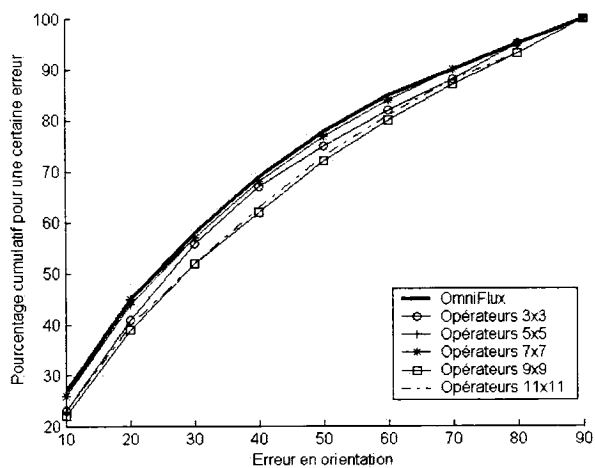


FIG. 3.49 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels

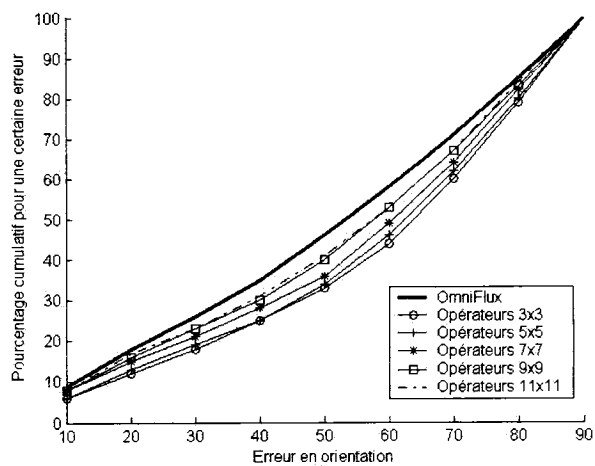


FIG. 3.50 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 50 et 70 pixels.

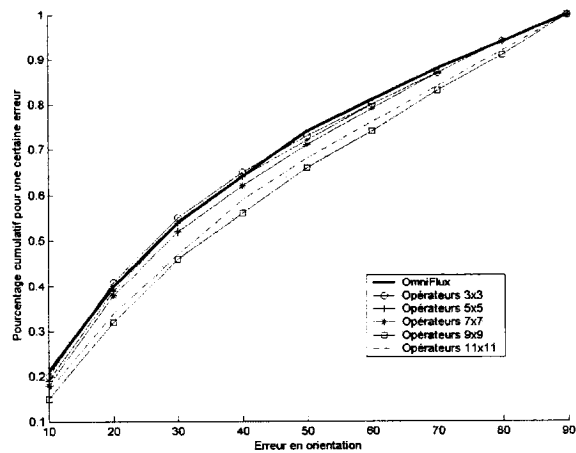


FIG. 3.51 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 110 et 130 pixels

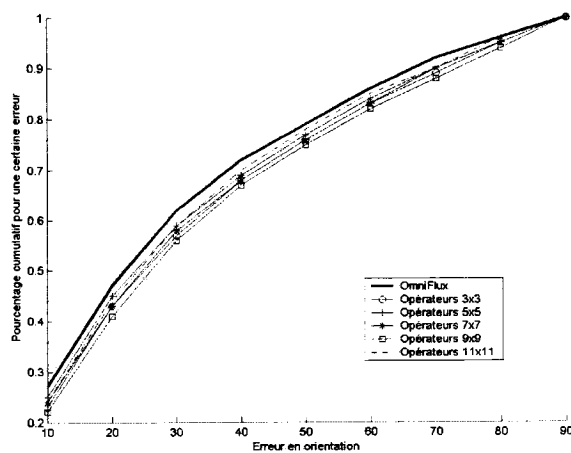


FIG. 3.52 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels

Les figures 3.48 et 3.49 présentent les diagrammes pour une translation perpendiculaire à l'axe de la caméra. Les figures 3.50 à 3.52 présentent l'erreur cumulative pour la rotation autour de l'axe. Tout comme pour la translation dans l'axe de la caméra, ces diagrammes permettent d'apprécier l'amélioration apportée en adaptant les opérateurs dans des rayons intermédiaires où l'adaptation est réellement effective. L'algorithme est soit équivalent, soit le mieux placé dans les diagrammes.

3.8.2 Séquences d'images réelles

Nous avons aussi comparé les algorithmes sur les séquences d'images réelles décrites précédemment. Tout comme dans la section précédente nous allons illustrer l'équivalence ou la supériorité de notre algorithme dans des couronnes où l'adaptation des opérateurs a lieu.

Dans le cas de la translation perpendiculaire à l'axe, les figures 3.53, 3.54 et 3.55 montrent les diagrammes cumulatifs pour différentes couronnes. La différence entre les algorithmes est moins flagrante mais le notre reste bien placé. Par exemple, pour la figure 3.54, les résultats sont similaires. Néanmoins dans les figures 3.53 et 3.54, notre algorithme est le plus en amont.

Dans le cas de la translation dans l'axe, les figures 3.56 et 3.57 montrent que notre algorithme n'est pas en amont mais il reste dans les trois meilleurs algorithmes.

Finalement, dans le cas de la rotation autour de l'axe, les figures 3.58 et 3.59 illustrent la position de notre algorithme. Il est très bien placé dans la figure 3.58 et en troisième position dans la figure 3.59.

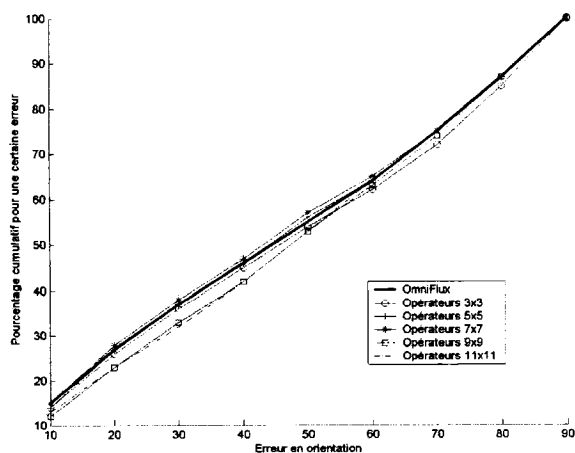


FIG. 3.53 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels dans le cas d'images réelles.

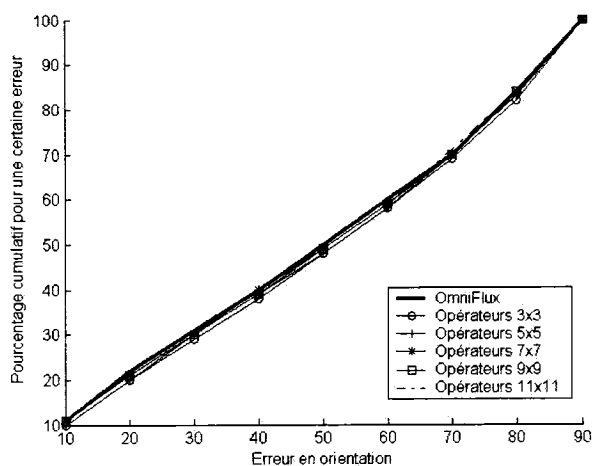


FIG. 3.54 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 90 et 110 pixels dans le cas d'images réelles.

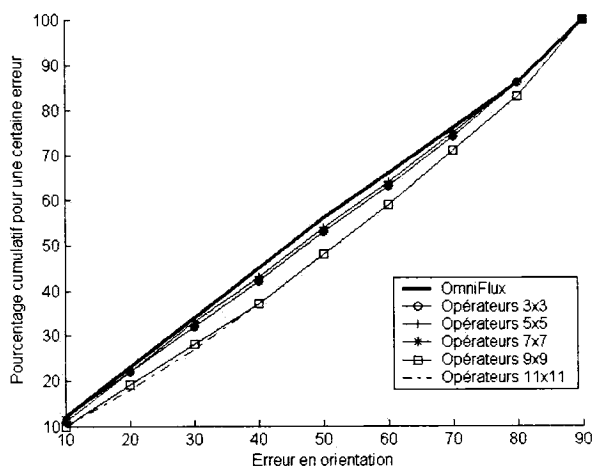


FIG. 3.55 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels dans le cas d'images réelles.

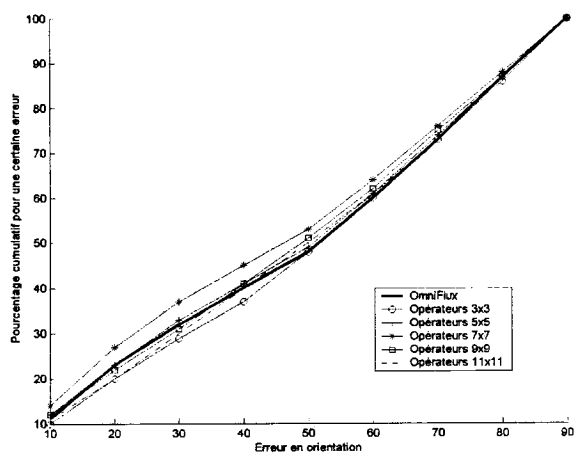


FIG. 3.56 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 50 et 70 pixels dans le cas d'images réelles.

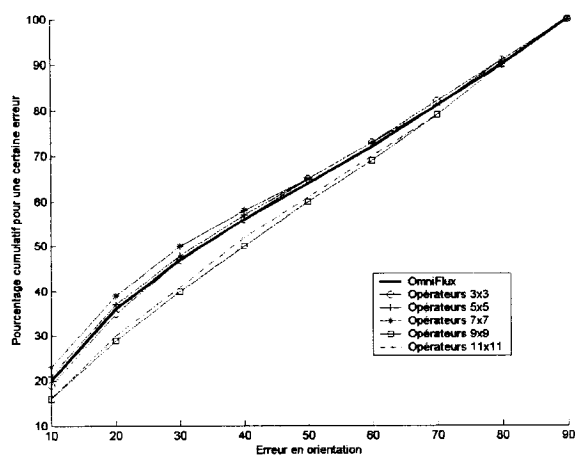


FIG. 3.57 Translation dans l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels dans le cas d'images réelles.

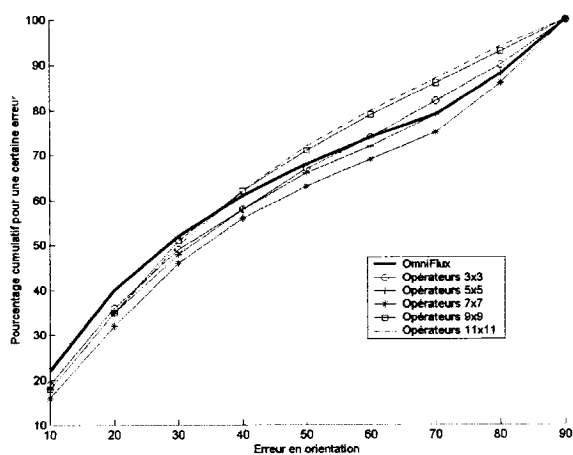


FIG. 3.58 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 30 et 50 pixels dans le cas d'images réelles.

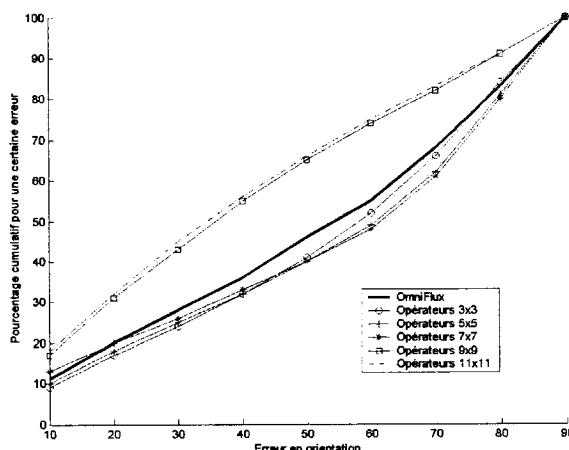


FIG. 3.59 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative dans la couronne comprise entre 130 et 150 pixels dans le cas d'images réelles.

3.8.3 Conclusion sur la comparaison entre algorithmes

Pour réaliser cette comparaison entre les algorithmes, nous avons tracé les diagrammes d'erreur cumulative dans le cas des trois mouvements étudiés pour les images synthétiques de la *configuration 1* et pour la série d'images réelles décrites en section 3.6.3. Ces diagrammes permettent de mettre en évidence que contrairement aux algorithmes utilisant des opérateurs homogènes, le notre profite de l'adaptation des opérateurs pour rester en amont des diagrammes. Il est donc soit dans la meilleure position, soit très bien placé.

Les études par couronne ont montré que ce sont dans les rayons intermédiaires, là où les adaptations d'opérateurs sont effectives, que notre algorithme se plaçait en très bonne ou en meilleure position.

Notons tout de même que l'écart entre les algorithmes est faible et qu'il ne concerne que l'orientation. Une étude sur l'amplitude du flux serait à ajouter indéniablement à notre étude si nous avions eu plus de temps.

3.9 Comparaison avec d'autres algorithmes

Pour juger de l'efficacité de l'adaptation des opérateurs, il est intéressant de l'utiliser pour d'autres algorithmes de calcul de flux optique existants, comme la technique différentielle dite de Horn et Schunck (Horn, 1981). Comme la méthode d'adaptation développée dans ce mémoire intervient dans le calcul des dérivées spatiales dans l'image, un nouvel algorithme a été codé pour la technique différentielle de Horn et Schunck en adaptant les opérateurs de dérivations.

Dans les figures suivantes «LK» désignera la technique différentielle attribuée à Lucas et Kanade et «LKOmn», son adaptation aux images omnidirectionnelles (la technique conventionnelle utilise des opérateurs de taille 3 par 3 et est issue de la librairie de fonctions OPENCV). De même, la seconde méthode différentielle sera notée «HS» et son adaptation, «HSOmn» (là aussi la technique «HS» est issue de la librairie de fonctions OPENCV).

Nous pouvons illustrer notre dernière technique par la figure 3.60 qui représente les vecteurs de flux calculés avec la méthode «HSOmn» dans le cas d'une image synthétique de la *configuration 1* pour une translation dans l'axe de la caméra de 1cm. Deux cas d'images synthétiques de la *configuration 1* et dans un cas d'image réelle ont permis de comparer les erreurs d'orientation du flux pour ces quatre algorithmes. Les figures 3.61, 3.62 et 3.63 en représentent les diagrammes d'erreur cumulative. La technique d'adaptation aux images omnidirectionnelles améliore la technique conventionnelle pour les images synthétiques (figures 3.61 et 3.62), mais pour l'image réelle (figure 3.63), il n'y a pas réellement de différence.

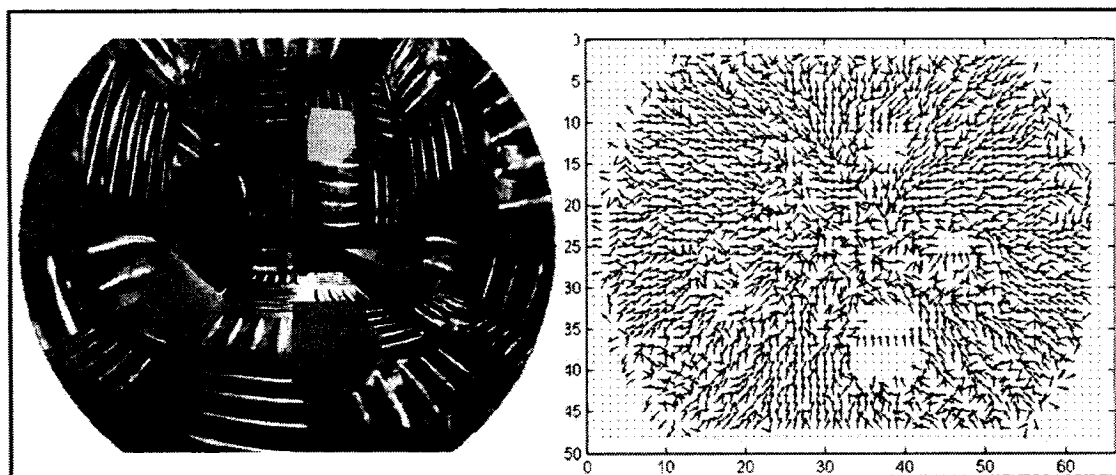


FIG. 3.60 HSOmni : représentation du flux optique pour une translation dans l'axe de la caméra pour une image synthétique. Un vecteur flux sur 10 est montré et chaque vecteur est normalisé.

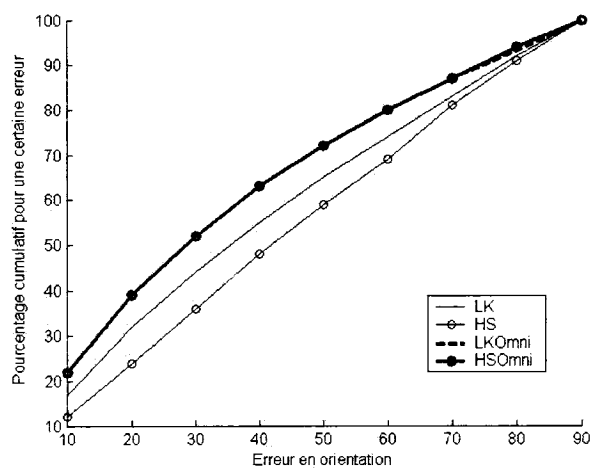


FIG. 3.61 Translation perpendiculaire à l'axe de la caméra : diagramme d'erreur cumulative.

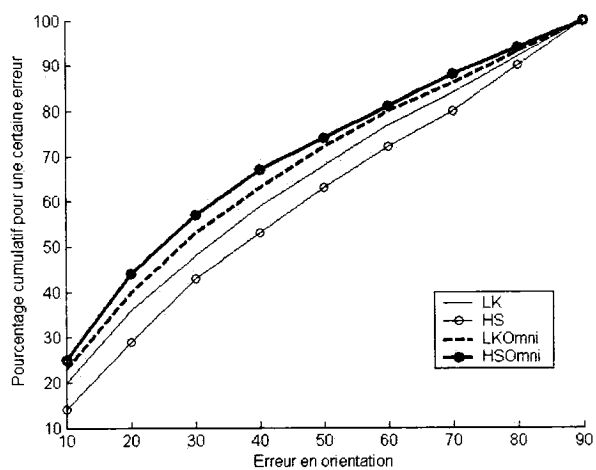


FIG. 3.62 Translation dans l'axe de la caméra : diagramme d'erreur cumulative.

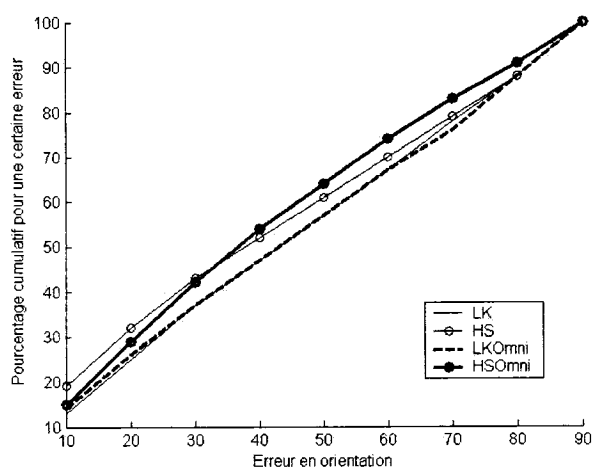


FIG. 3.63 Rotation autour de l'axe de la caméra : diagramme d'erreur cumulative.

3.10 Conclusion sur l'adaptation des opérateurs pour le calcul de flux optique dans les images omnidirectionnelles

La technique de calcul de flux optique choisie pour ces travaux a été une technique différentielle dite de Lucas et Kanade, déjà développée dans des programmes de nombreuses manières par le passé. Cette technique et plus précisément les opérateurs de dérivation qu'elle utilise ont été adaptés pour qu'ils soient plus cohérents avec la géométrie projective des caméras omnidirectionnelles en conservant tous les calculs sur le plan image. La validité de l'algorithme a été étudiée à l'aide d'images synthétiques, certaines simples comme des voisinages gaussiens se déplaçant sur le plan image, d'autres plus complexes qui recréent un environnement 3D vue par une caméra omnidirectionnelle virtuelle de même dimension que celle réelle utilisée au laboratoire. Cette dernière a d'ailleurs servi à prendre des images qui ont permis de valider en orientation le calcul de flux.

Les approximations des opérateurs «omnidirectionnels» ont fourni des résultats dont les erreurs sont légèrement inférieures à celles fournies par des opérateurs «conventionnels» pour une série d'images synthétiques créées. En ce qui concerne les séquences d'images réelles obtenues au laboratoire, les résultats restent meilleurs mais pas nécessairement de façon significative pour toutes les séquences.

L'adaptation d'opérateurs a été essayée sur une autre technique différentielle de calcul de flux. Les résultats sont encourageants et permettent d'espérer une adaptation d'autres algorithmes connus de calcul de flux.

Une composante manque cruellement à cette étude et c'est l'étude de l'amplitude du flux optique dans certaines séquences d'images. Pour ce qui est des images synthétiques et réelles, une validation basée sur l'amplitude n'a pas été réalisée. Heureusement, les formes simples ont permis de valider notre algorithme et l'étude des erreurs en orientation a permis d'effectuer une partie de sa validation et de le comparer à d'autres.

Une étude future pourrait aussi comporter des séquences de plus de deux images et permettrait compléter les résultats.

Le second point est que les algorithmes utilisés n'ont pas été écrits dans un souci d'optimisation. Ils manipulent des opérateurs de grandes tailles (11 par 11 au maximum) qui sont chers en temps de calcul. Une des études futures pourrait estimer la diminution des erreurs obtenue avec l'adaptation des opérateurs, en fonction de la vitesse de calcul. Si le nombre d'articles proposant un calcul du flux optique adapté aux caméras omnidirectionnelles est faible, c'est que les groupes de recherche acceptent l'erreur commise en gardant les opérateurs utilisés avec des images en perspective, dans le but d'optimiser la vitesse de leur système.

Le calcul du flux optique ayant été détaillé, le chapitre suivant présente le protocole de détection de mouvement.

CHAPITRE 4

SYSTÈME VISUEL ET PROTOCOLE DE DÉTECTION DE MOUVEMENT

4.1 Système visuel

L'objectif principal de ce projet est la mise en place d'un système de vision associant une système catadioptrique, avec un grand champ de vision mais une résolution variable en charge d'une détection de mouvement, et une vision directionnelle, chargée de récolter à plus grande résolution des informations sur les zones détectées. Les composantes de notre «Système Visuel» peuvent être classées en deux catégories :

- celles reliées à l'architecture matérielle du système, comme le placement des caméras ou le choix de la plate forme mobile,
- celles reliées à l'architecture logicielle du système comme le langage de programmation, l'architecture ou les librairies utilisées.

4.1.1 Architecture matérielle

Puisque la partie centrale d'une image omnidirectionnelle est occupée par le reflet de la caméra, l'idée était de placer une seconde caméra au dos de la caméra omnidirectionnelle pour obtenir les pixels «manquants», les axes optiques des deux caméras se confondant avec l'axe du miroir hyperboloïdal. La figure 4.1 représente le système construit pendant le projet. Pour atteindre notre objectif principal, construire un système de vision détectant les mouvements indépendants, il faut pouvoir tester ce système en imposant des mouvements à la caméra. Pour ce faire

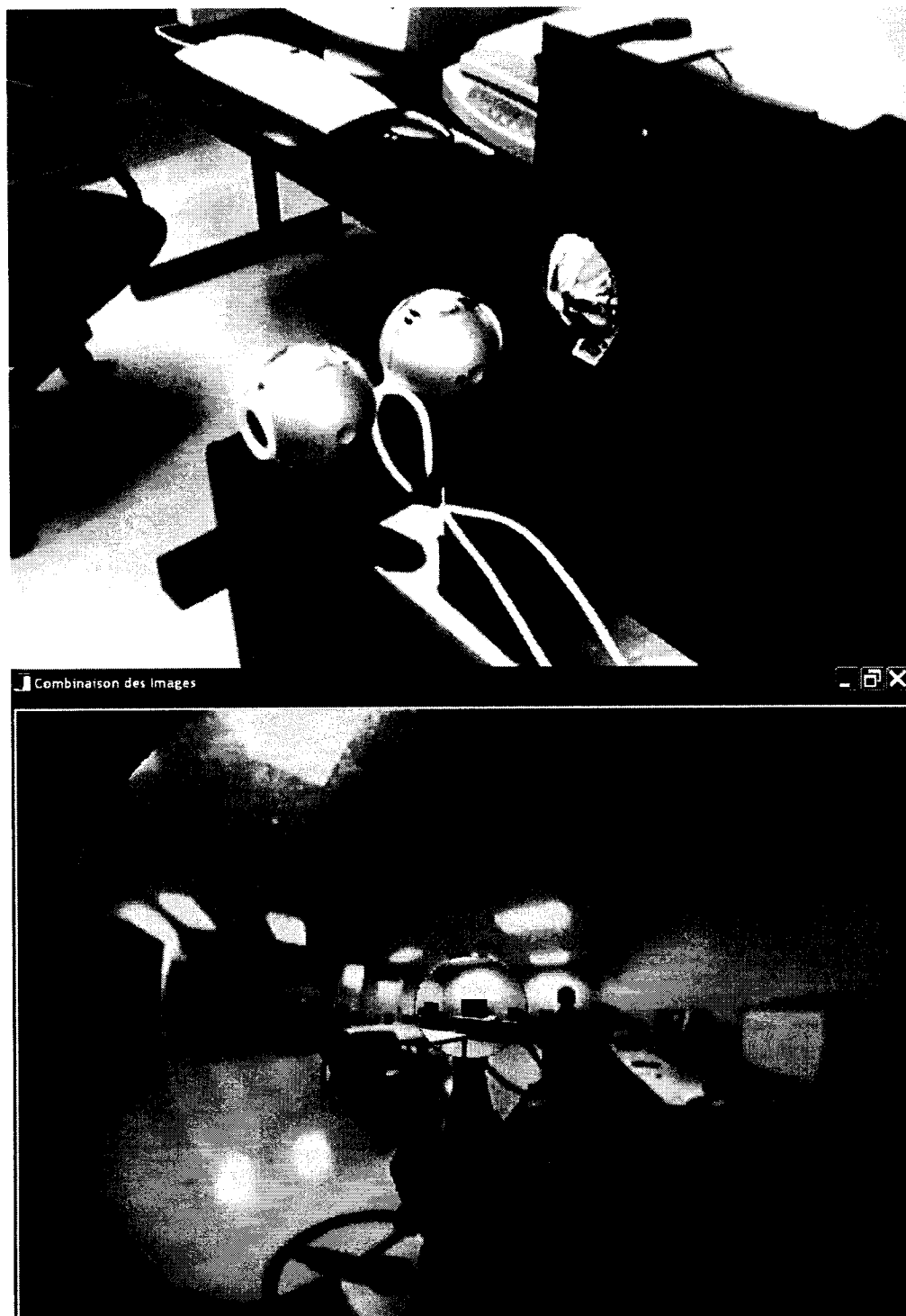


FIG. 4.1 Système utilisé : deux caméras et un miroir hyperboloïdal posés sur un moteur PAN-TILT. La deuxième figure montre les deux visions imbriquées, le reflet de la caméra a été remplacé par l'image fournie par le seconde caméra.

le système de caméras est monté sur deux moteurs qui permettent deux rotations (voir figure 4.1). Par la suite, le mouvement de lacet sera appelé PAN, et le mouvement de tangage sera appelé TILT. Les deux moteurs permettent de fournir un déplacement connu et une vitesse instantanée connue aux caméras.

4.1.2 Architecture logicielle

Tous les programmes ont été écrits en C++ et utilisent certaines bibliothèques de fonctions.

- La bibliothèque de vision OPENCV pour traiter les images a été utilisée. Elle présente un format d'image très complet pour traiter toute la partie vision de l'étude ainsi que des fonctions optimisées de traitement d'image.
- Le logiciel d'acquisition d'images a été écrit au GRPR.
- La bibliothèque pour la commande des moteurs a été écrite au GRPR.
- Toutes ces composantes ont été réunies dans une même architecture, et dans notre cas une interface graphique a été codée grâce à la bibliothèque wxWidget, pour rendre le programme utilisable par tout utilisateur. Dans le futur, cette architecture étant destinée à être installée sur une plate forme mobile, il fallait pouvoir porter tout le code sur Linux. wxWidget permet de coder des processus légers, ce qui est essentiel pour pouvoir dans le même temps afficher les images, les traiter et diriger les moteurs.

Un guide d'utilisation de l'interface est fourni en annexe III.

4.2 Protocole de détection

En se basant sur la discussion de la section 1.2.1, pour pouvoir adapter les méthodes connues de segmentation dans une image, il faut pouvoir effectuer les deux étapes

suivantes :

- exprimer un modèle de mouvement simple dans le cas d’une image omnidirectionnelle,
- relier le flux optique (calculable à partir des images) au modèle de mouvement.

4.2.1 Modèle de mouvement pour une caméra omnidirectionnelle

La complexité du modèle de mouvement à exprimer, dans le cas omnidirectionnel, a forcé les chercheurs à contourner le problème en utilisant les projections sphériques, et notamment celle de la théorie unificatrice de (Daniilidis, 2000).

Soit un point P fixe de l’espace, de coordonnées (X, Y, Z) dans le repère de la caméra, centré au centre de la sphère O et dont l’axe Z coïncide avec l’axe du miroir, et soit un mouvement subi par la caméra, une rotation R et une translation T , la vitesse instantanée de P s’exprime alors :

$$\dot{P} = -T - R \times P \quad (4.1)$$

Nous considérons maintenant le point \hat{P} sur la sphère, avec $\hat{P} = \frac{P}{\|P\|}$.

Le champ de mouvement s’écrit alors :

$$U(\hat{P}) = \frac{1}{\|P\|}((T \cdot \hat{P}) \cdot \hat{P} - T) - R \times \hat{P} \quad (4.2)$$

Pour éliminer la norme de P , il faut effectuer le produit vectoriel de (4.2) par \hat{P} :

$$\hat{P} \times U(\hat{P}) = \frac{1}{\|P\|}(-\hat{P} \times T) - \hat{P} \times (R \times \hat{P}) \quad (4.3)$$

puis le produit scalaire par T en admettant que $T \neq 0$,

$$T \cdot (\hat{P} \times U(\hat{P})) = -T \cdot (\hat{P} \times (R \times \hat{P})) \quad (4.4)$$

ou encore :

$$T.(\hat{P} \times (U(\hat{P}) + (R \times \hat{P}))) = 0 \quad (4.5)$$

Il est possible de retrouver que $U(\hat{P})$ est tangent à la sphère :

$$U(\hat{P}).\hat{P} = \frac{1}{\|\hat{P}\|}((T.\hat{P}).\hat{P}.\hat{P} - T.\hat{P}) - (R \times \hat{P}).\hat{P}$$

or $\|\hat{P}\| = 1$, donc

$$U(\hat{P}).\hat{P} = \frac{1}{\|\hat{P}\|}((T.\hat{P}).\|\hat{P}\| - T.\hat{P}) = 0$$

En reprenant les notations de la partie 2.3.2, en posant $U(\hat{P}) = (u_1, u_2, u_3)$, et en désignant par A , B et C les vitesses des rotations du miroir selon les axes X , Y et Z respectivement (voir figure 4.2), le système d'équations s'écrit :

$$\begin{aligned} U(\hat{Y}u_3 - \hat{Z}u_2) &= -U[\hat{Y}(A\hat{Y} - B\hat{X}) - \hat{Z}(C\hat{X} - A\hat{Z})] \\ V(\hat{Z}u_1 - \hat{X}u_3) &= -V[\hat{Z}(B\hat{Z} - C\hat{Y}) - \hat{X}(A\hat{Y} - B\hat{X})] \\ W(\hat{X}u_2 - \hat{Y}u_1) &= -W[\hat{X}(C\hat{X} - A\hat{Z}) - \hat{Y}(B\hat{Z} - C\hat{Y})] \end{aligned} \quad (4.8)$$

Dans le cas $T = 0$, l'équation 4.2 devient tout simplement :

$$U(\hat{P}) = -R \times \hat{P} \quad (4.9)$$

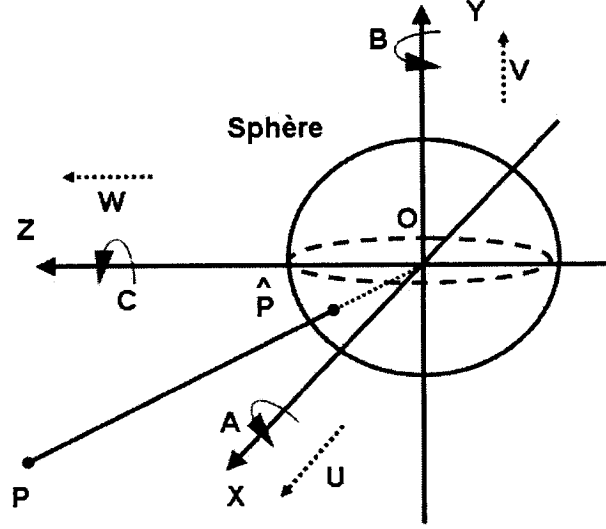


FIG. 4.2 Figure illustrant les différentes rotations et les translations de la sphère, ainsi que le point projeté sur la sphère.

et le système d'équations s'écrit alors :

$$\begin{aligned}
 u_1 &= B\hat{Z} - C\hat{Y} \\
 u_2 &= C\hat{X} - A\hat{Z} \\
 u_3 &= A\hat{Y} - B\hat{X}
 \end{aligned}
 \tag{4.10}$$

4.2.2 Relier le flux optique au modèle de mouvement

En regardant le système 4.8, pour relier le champ de vitesse dans l'image, ou flux optique, $(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t})$, au champ de vitesse des points sur la sphère, $(\frac{\partial \hat{X}}{\partial t}, \frac{\partial \hat{Y}}{\partial t}, \frac{\partial \hat{Z}}{\partial t})$, nous pouvons utiliser le Jacobien J . Il est calculé dans la section 2.3.2 et permet de relier

le flux optique à la vitesse des points sur la sphère :

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \frac{\partial \hat{X}}{\partial t} \\ \frac{\partial \hat{Y}}{\partial t} \\ \frac{\partial \hat{Z}}{\partial t} \end{pmatrix} = J \begin{pmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{pmatrix} \quad (4.11)$$

Il est désormais possible de calculer $U(\hat{P})$ ou encore ses trois composantes u_1 , u_2 et u_3 à partir des vecteurs du flux optique, et nous avons ainsi relié le flux au modèle de mouvement.

4.2.3 Schéma du protocole

Grâce à la section 4.2.1, nous pouvons dire que si un point de l'espace P est fixe, et si la caméra est en déplacement, alors sa projection sur la sphère \hat{P} possède des coordonnées de position, $(\hat{X}, \hat{Y}, \hat{Z})$, et de vitesse, (u_1, u_2, u_3) , qui respectent le système 4.8.

Par contre si le point est la projection d'une partie de la scène en déplacement indépendant, le champ de vitesse u_1, u_2, u_3 ne respecte plus le système 4.8.

Comme énoncé dans les objectifs de ce projet (1.2), c'est sur la détection de mouvement que notre étude a porté. Plus particulièrement, c'est la différenciation entre la représentation dans l'image du mouvement propre de la caméra et la représentation dans l'image des zones correspondant à des mouvements indépendants.

Notre protocole s'appuie sur la connaissance des paramètres (U, V, W) pour la translation et A, B, C pour la rotation (dans la pratique, cela va correspondre au renvoi par le moteur des vitesses de rotation PAN et TILT).

Pour détecter le mouvement, la procédure suivante sera donc employée :

- Pour un point $P(x, y)$ sur le plan image, calcul du point $\hat{P} = (\hat{X}, \hat{Y}, \hat{Z})$ corres-

pendant sur la sphère.

- Calcul en ce point P , du flux optique sur la sphère à partir du flux optique calculé sur le plan image au point P .
- Analyse du respect ou non respect du système 4.8.

Comme nous l'avons dit à la fin du chapitre précédent, l'optimisation du calcul de flux optique n'a pas pu être réalisée. De plus il manque un axe important à son étude, c'est l'étude en amplitude du calcul de flux avec les opérateurs adaptés. Nous avons donc choisi d'utiliser pour le calcul de flux optique, l'algorithme disponible dans la librairie OPENCV. Cette décision a aussi été prise en s'appuyant sur les articles (Sastry, 2003), (Daniilidis, 2002) et (Gandhi, 2003), qui reconnaissent la possibilité d'effectuer un calcul plus juste du flux optique dans une image omnidirectionnelle, mais qui recommande l'emploi de techniques plus traditionnelles pour toute les applications temps réelles.

4.2.4 Particularités du montage et changements de repère

Il est important de pouvoir calculer les expressions des rotations et translations effectuées par le système en fonction des vitesses de rotations des deux moteurs PAN et TILT. Les vitesses instantanées des points de l'espace doivent être exprimées en fonction des vitesses de rotation des deux moteurs. Deux repères seront utilisés : le repère $(R, \vec{i}, \vec{j}, \vec{k})$ avant les rotations et le repère $(R', \vec{i}', \vec{j}', \vec{k}')$ après les rotations (voir figure 4.3). Notons (X, Y, Z) les coordonnées d'un point de l'espace dans le repère du système sans rotation. Ce système est centré sur la sphère virtuelle correspondant au miroir utilisé dans le cadre de la théorie unificatrice. C'est-à-dire

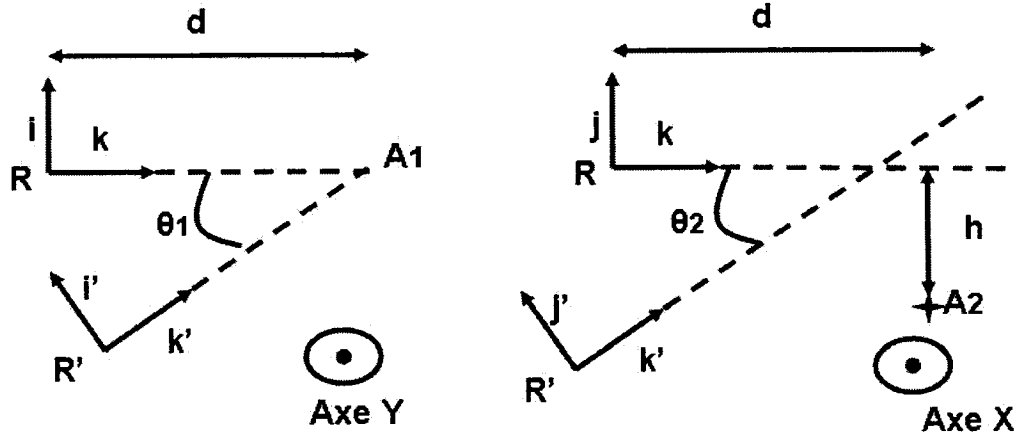


FIG. 4.3 Projection en 2D des rotations PAN et TILT du système, A_i étant la position du centre de la rotation et R le centre du repère lié au miroir.

que les paramètres l et m exprimés dans la section 2.3.1 sont calculés :

$$l = 0.9662 \quad (4.12)$$

$$m = -0.9474 \quad (4.13)$$

$$(4.14)$$

Notons (X', Y', Z') les coordonnées du même point dans le repère du système après les deux rotations. θ_1 et θ_2 désigneront les angles de rotation PAN et TILT respectivement. Pour la rotation PAN du moteur, l'axe de rotation passe par A_1 (voir figure 4.4) et les équations de changement de repère sont :

$$X' = d \sin \theta_1 - Z \sin \theta_1 + X \cos \theta_1 \quad (4.15)$$

$$Z' = d - d \cos \theta_1 + Z \cos \theta_1 + X \sin \theta_1$$

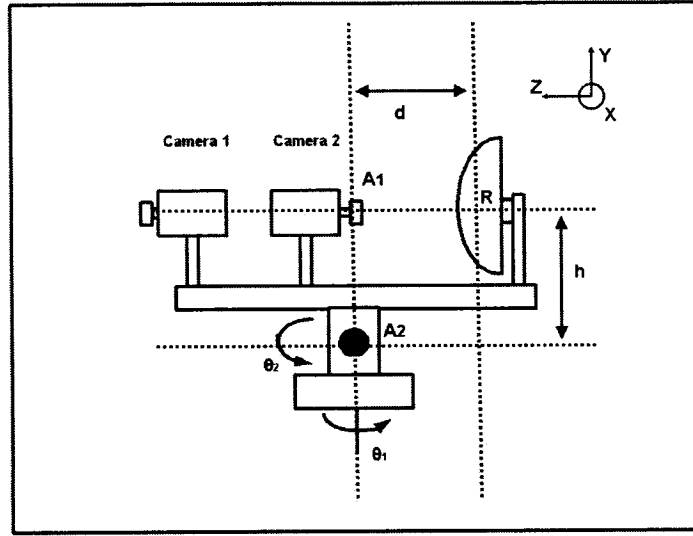


FIG. 4.4 Système visuel.

De la même façon, avec le mouvement TILT et A_2 comme axe de rotation :

$$Y' = -h + d \sin \theta_2 + h \cos \theta_2 - Z \sin \theta_2 + Y \cos \theta_2 \quad (4.16)$$

$$Z' = d - d \cos \theta_2 + h \sin \theta_2 + Z \cos \theta_2 + Y \sin \theta_2$$

Finalement avec les deux rotations, les équations de changement de repère sont :

$$\begin{aligned} X' &= (d - Z) \cos \theta_2 \sin \theta_1 - (Y + h) \sin \theta_1 \sin \theta_2 + X \cos \theta_1 \\ Y' &= -h + (d - Z) \cos \theta_1 \sin \theta_2 \\ &\quad - X \sin \theta_1 \sin \theta_2 + (h + Y) \cos \theta_2 \\ Z' &= d + (Z - d) \cos \theta_1 \cos \theta_2 + X \cos \theta_2 \sin \theta_1 + (h + Y) \sin \theta_2 \end{aligned} \quad (4.17)$$

Les vitesses instantanées sont donc :

$$\begin{aligned}
 \dot{X} &= (d - Z)(-\dot{\theta}_2 \sin \theta_2 \sin \theta_1 + \dot{\theta}_1 \cos \theta_2 \cos \theta_1) \\
 &\quad - (Y + h)(\dot{\theta}_1 \cos \theta_1 \sin \theta_2 + \dot{\theta}_2 \cos \theta_2 \sin \theta_1) - X \dot{\theta}_1 \sin \theta_1 \\
 \dot{Y} &= (d - Z)(-\dot{\theta}_1 \sin \theta_1 \sin \theta_2 + \dot{\theta}_2 \cos \theta_1 \cos \theta_2) \\
 &\quad - X(\dot{\theta}_1 \cos \theta_1 \sin \theta_2 + \dot{\theta}_2 \cos \theta_2 \sin \theta_1) - (h + Y)\dot{\theta}_2 \sin \theta_2 \\
 \dot{Z} &= (d - Z)(\dot{\theta}_1 \sin \theta_1 \cos \theta_2 + \dot{\theta}_2 \sin \theta_2 \cos \theta_1) \\
 &\quad + X(\dot{\theta}_1 \cos \theta_1 \cos \theta_2 - \dot{\theta}_2 \sin \theta_1 \sin \theta_1) + (h + Y) \cos \theta_2 \dot{\theta}_2
 \end{aligned}$$

En utilisant des rotations infinitésimales :

$$\cos \theta_i \simeq 1$$

$$\sin \theta_i \simeq 0$$

(4.18)

Le système 4.18 devient :

$$\begin{aligned}
 \dot{X} &= (d - Z)\dot{\theta}_1 \\
 \dot{Y} &= (d - Z)\dot{\theta}_2 \\
 \dot{Z} &= (h + Y)\dot{\theta}_2 + X\dot{\theta}_1
 \end{aligned} \tag{4.19}$$

Pour notre système, en reprenant l'équation :

$$\dot{P} = -T - R \times P \tag{4.20}$$

et en la développant sur chacun des axes,

$$\begin{aligned}\dot{X} &= -U + B.Z - C.Y \\ \dot{Y} &= -V + C.X - A.Z \\ \dot{Z} &= -W + A.Y - B.X\end{aligned}$$

En identifiant ce dernier système avec le système 4.19 nous pouvons en déduire :

$$\begin{aligned}A &= -\dot{\theta}_2 \\ B &= \dot{\theta}_1 \\ C &= 0 \\ U &= -d\dot{\theta}_1 \\ V &= -d\dot{\theta}_2 \\ W &= -h\dot{\theta}_2\end{aligned}\tag{4.21}$$

4.2.5 Système d'équations dans le cadre des moteurs PAN-TILT

Nous pouvons aussi simplifier le système 4.8 dans le cadre des mouvements PAN et TILT permis par le système.

4.2.5.1 Mouvement PAN

Dans le cas d'un mouvement PAN, le système 4.21 fournit $V = 0$, $W = 0$, $A = 0$, $B \neq 0$, $C = 0$ et $U \neq 0$, et le système 4.8 se réduit à une équation :

$$\hat{Y}u_3 - \hat{Z}u_2 = B\hat{X}\hat{Y} \quad (4.22)$$

4.2.5.2 Mouvement TILT

Dans le cas d'un mouvement TILT, le système 4.21 fournit $U = 0$, $V \neq 0$, $W \neq 0$, $A \neq 0$, $C = 0$ et $B = 0$, et le système 4.8 se réduit à deux équations :

$$\begin{aligned} \hat{Z}u_1 - \hat{X}u_3 &= A\hat{X}\hat{Y} \\ \hat{X}u_2 - \hat{Y}u_1 &= A\hat{X}\hat{Z} \end{aligned} \quad (4.23)$$

4.3 Seuil de détection

La détection se base sur le respect du système 4.8 en tout point de l'image.

Pour un mouvement PAN, le système est donné dans système 4.22. Il faut alors faire la différence entre le terme $G = \hat{Y}u_3 - \hat{Z}u_2$ calculé à partir du flux optique dans l'image, et le terme $D = B\hat{X}\hat{Y}$ calculé à partir de la vitesse de rotation PAN connue. Un seuil ϵ constant est fixé et le système détecte un mouvement si la condition suivante est respectée :

$$|G - D| > \epsilon \quad (4.24)$$

Pour le mouvement TILT, il y a deux équations :

$$\begin{aligned}\hat{Z}u_1 - \hat{X}u_3 &= A\hat{X}\hat{Y} \\ \hat{X}u_2 - \hat{Y}u_1 &= A\hat{X}\hat{Z}\end{aligned}\tag{4.25}$$

Avec $G_1 = \hat{Z}u_1 - \hat{X}u_3$, $D_1 = A\hat{X}\hat{Y}$ et $G_2 = \hat{X}u_2 - \hat{Y}u_1$, $D_2 = A\hat{X}\hat{Z}$, le système détecte un mouvement indépendant si

$$|G_1 - D_1| > \epsilon \quad \text{ou} \quad |G_2 - D_2| > \epsilon \tag{4.26}$$

Grâce aux premiers tests, il a été possible de constater que si la valeur pour ϵ est choisie convenablement, le protocole réalise une détection des mouvements indépendants. Nous n'avons pas choisi de mettre en place une détection de seuil automatique, mais comme les images sont bruitées et que la luminosité des scènes influence le calcul du flux donc le protocole de détection, nous avons intégré au programme un sélecteur pour faire varier la valeur de ϵ et observer les changements en temps réel.

La figure 4.5 illustre cette détection pour un opérateur qui déplace son bras. Le protocole mérite d'être amélioré car des erreurs de détection peuvent survenir. Par exemple, certaines arêtes dans l'image présentent une norme de flux optique importante et satisfont la condition 4.24, mais ne constituent pas une zone en mouvement indépendant. Ainsi, dans la figure 4.6, le piéton qui se déplace est bien détecté mais plusieurs points dans l'environnement statique sont également considérés comme en mouvement.

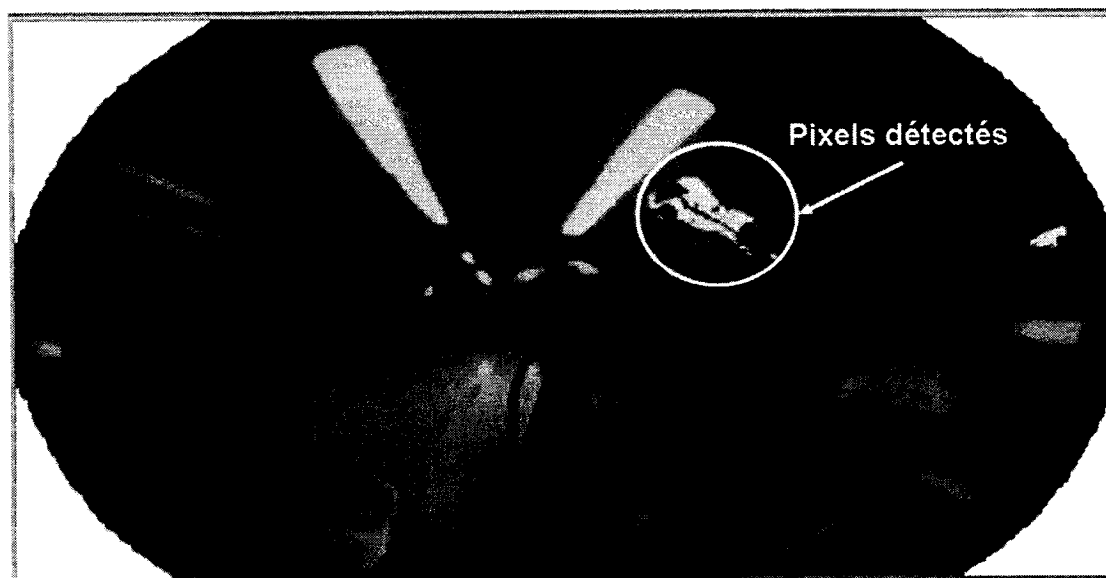


FIG. 4.5 Détection pixel par pixel : l'opérateur bouge le bras et des pixels en mouvement indépendant sont détectés.

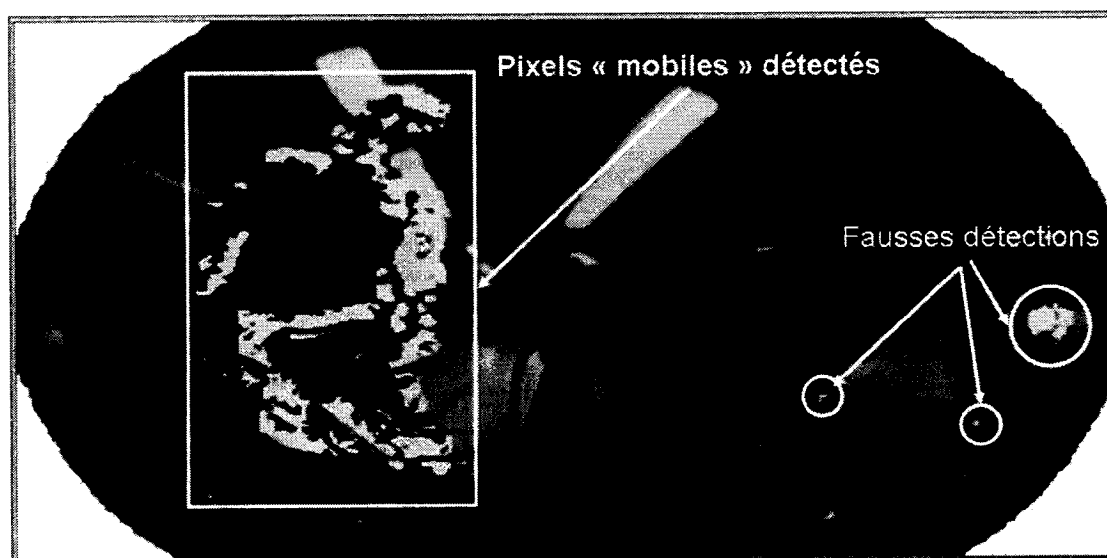


FIG. 4.6 Détection pixel par pixel : pixels en mouvement détectés et fausses détections sont présentes.

4.4 Création de zones en mouvement indépendant et suivi

Une fois que chaque pixel s'est vu attribué une étiquette «en mouvement» ou «immobile», il s'agit de regrouper ces pixels pour former des zones de pixels «en mouvement».

La technique utilisée est la fusion de région qui se déroule en deux étapes et utilise un seuil de taille λ :

1. pour chaque pixel p considéré comme «mobile», les pixels de son voisinage sont examinés. Si un pixel voisin p' est aussi «mobile», alors on assigne p' et p à la même zone, et la procédure est recommencée pour le pixel p' du voisinage. Les itérations sont arrêtées quand tous les pixels ont été examinés ou attribués à une zone.
2. Une fois toutes les zones identifiées, les tailles des zones sont observées et doivent être supérieure à λ . Les tailles des zones correspondent au nombre de pixels dans cette zone. Cela permet d'éliminer les zones de taille trop faible qui correspondent à des fausses détections ou à un mauvais calcul de flux ciblé sur un pixel.

Il s'agit ensuite d'algorithmes de suivi standards qui consistent à créer une liste dans laquelle sont stockés toutes les régions détectées ainsi que leurs principales caractéristiques (aire, position, intensité).

Une dernière caractéristique est ajoutée et c'est le nombre de fois où la zone a été détectée que nous nommerons NbD . Notre détection s'effectue à chaque image et tente d'attribuer à chaque nouvelle région en mouvement détectée, une région déjà présente dans l'image précédente en utilisant des critères de distance. La position dans l'image du barycentre de la zone détectée B est comparée à la position dans l'image de toutes les zones classées dans la liste. Si la distance entre B et un barycentre de la liste B' est inférieure à 25 pixels, les caractéristiques de B' deviennent

celles de B et son NbD augmente de 1 (la valeur 25 est obtenue par apprentissage après avoir effectué de nombreux tests sur des personnes en déplacement devant la caméra). Si une zone de la liste ne s'est vue affecter aucune nouvelle zone, son NbD diminue de 1. Quand NbD arrive à 0, la zone est éliminée de la liste. Cela permet en 5 images d'éliminer les «fausses» zones qui ne sont apparues que pendant une image.

Si la région ne peut être mise en correspondance, elle est considérée comme une nouvelle région et elle est ajoutée à la liste des régions en mouvements déjà existantes.

Maintenant que le système physique et le protocole de détection ont été présentés, la partie suivante décrit les résultats obtenus pour différentes séquences d'images.

4.5 Résultats

Le protocole détaillé précédemment a été testé successivement sur des séquences d'images synthétiques et des séquences d'images réelles.

Les caméras utilisées pour les séquences d'images synthétiques ou réelles prennent 30 images par seconde.

4.5.1 Analyse de séquences d'images synthétiques

4.5.1.1 Création de la scène

Pour effectuer une analyse précise du protocole de détection, un environnement synthétique a été créé grâce au logiciel 3D Studio Max.

La première étape consiste à créer une surface réfléchissante identique au miroir

hyperboloïdal. C'est la même surface réfléchissante qui a été utilisée pour créer des séquences d'images pour tester les algorithmes de flux optique dans le chapitre 3. La seconde étape consiste à créer une salle pour symboliser un environnement fermé.

La troisième étape consiste à créer des objets pour les faire évoluer dans la scène. Enfin il faut paramétrer tous les mouvements des objets et de la caméra. La scène comprend un arrière plan fixe ainsi qu'un certain nombre d'objets en mouvement :

- une sphère S de 80 cm de rayon,
- un cube C de dimensions [largeur, longueur, hauteur] = [100, 100, 100] (en cm),
- un pavé P de dimensions [largeur, longueur, hauteur] = [75, 50, 10] (en cm).

Différents éléments structuraux ont également été ajoutés à l'arrière plan. Pour plus de simplicité l'environnement virtuel peut être défini comme une salle cubique dans laquelle les objets bougent principalement dans le sens de la largeur devant la caméra, dans le sens de la hauteur lorsqu'ils s'élèvent ou descendent et finalement en profondeur lorsqu'ils se rapprochent ou qu'ils s'éloignent de la caméra. La caméra reproduit le mouvement PAN évoqué précédemment en effectuant des rotations de 90° dans un sens puis dans l'autre à la vitesse de $0,86^\circ.s^{-1}$.

4.5.1.2 Analyse des mouvements dans la scène synthétique

Une fois l'environnement créé et le système de vision reproduit de façon synthétique, il faut tester le protocole de détection. À chaque illustration des détections de mouvement, l'image en question sera affichée à gauche sur la figure et la même image avec les zones détectées et dessinées en blanc à droite sur la figure. L'analyse des résultats sera effectuée par objet.

Déplacement de la sphère :

- Durant les 100 premières images, elle se déplace en largeur de 2.7 m vers la gauche à $0.8 m.s^{-1}$. Son déplacement est bien détecté comme l'illustre la figure

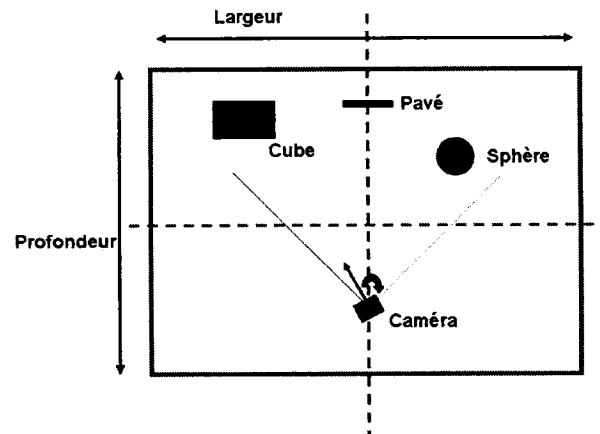


FIG. 4.7 Vue du dessus de la scène virtuelle.

4.9. Les parties détectées et regroupées en zone sont encadrées par un rectangle de taille adéquate.

- Entre les images 100 et 200, il arrive qu'elle soit détectée pendant deux ou trois images de suite alors qu'elle reste immobile. La différence de contraste entre la sphère et l'arrière-plan fournit une variation de flux optique trop grande pendant le mouvement de la caméra et fournit une mauvaise détection.
- Entre les images 200 et 300, elle avance vers la caméra de 70 cm à 0.21 m.s^{-1} . Elle est détectée par le système comme le montre l'image 4.10, puis sort du champ de la caméra.
- Entre les images 300 et 350, elle recule de 70 cm à la vitesse de 0.42 m.s^{-1} . Elle est là aussi bien détectée.
- Entre les images 350 et 400, elle se déplace en largeur de 2.7 m vers la droite à 0.8 m.s^{-1} . Ce mouvement rapide est propice à une bonne détection et un rectangle entoure la sphère comme le montre la figure 4.11.

La sphère a permis de tester à des vitesses moyennes plusieurs catégories de mouvement dans l'image qui n'ont pas tous le même impact sur le plan image.

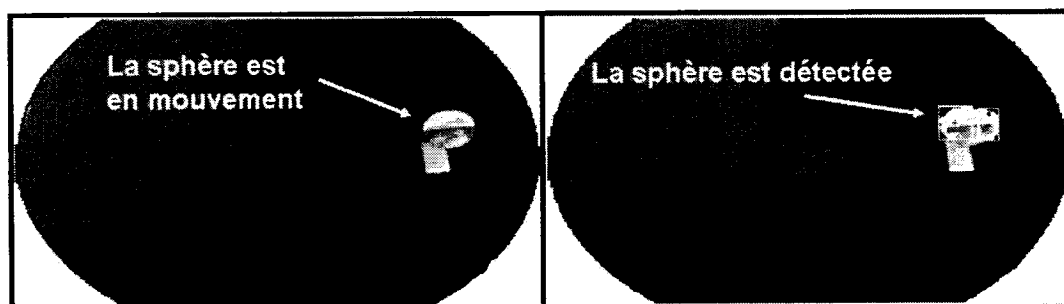


FIG. 4.8 Les 100 premières images : seule la sphère est en mouvement.

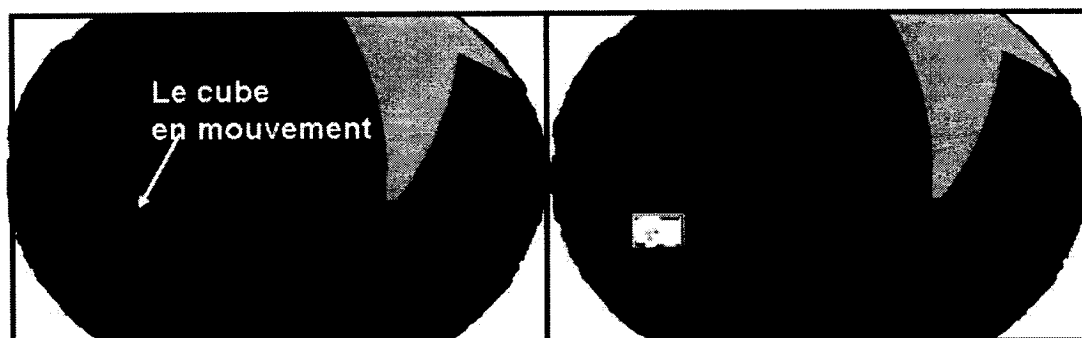


FIG. 4.9 Entre les images 100 et 200 : le cube est en mouvement et est détecté.

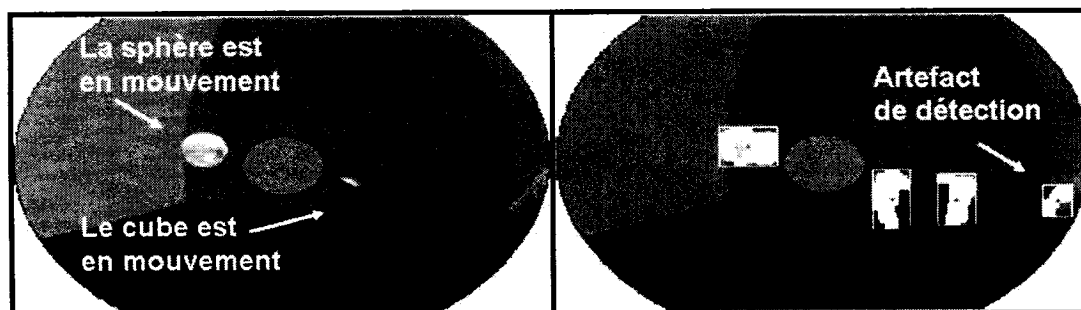


FIG. 4.10 Entre les images 200 et 300 : la sphère et le cube sont en mouvement et sont détectés. Une fausse détection est aussi présente.

Déplacement du cube :

- Durant les 100 premières images, le cube reste immobile mais comme pour la sphère, il arrive que le programme détecte un mouvement inexistant. Comme précédemment, cela s'explique par la différence de contraste entre l'objet et l'arrière-plan qui fournit de trop grandes valeurs de flux. Notons que cette fausse détection est uniquement ponctuelle.
- Entre les images 100 et 200, le cube se déplace rapidement vers la droite de la scène (2.16 m.s^{-1}). Cela est bien détecté par le système.
- Entre les images 200 et 300, il parcourt le chemin inverse et est toujours détecté comme le montre l'image 4.10.
- Le dernier mouvement est un mouvement de va et vient en hauteur (1.8 m.s^{-1}) durant lequel le déplacement du cube est repéré par le système (voir figure 4.11).

Les déplacements du cube ont permis de tester de grands déplacements rapides. Il a permis, comme la sphère, de prouver que le système détecte bien les objets en mouvement indépendants par rapport à l'arrière plan, mais aussi de montrer que certaines mauvaises détections apparaissent dans des périodes d'immobilité. Le dernier objet en mouvement est un pavé.

Déplacement du pavé :

- Durant les 150 premières images, le pavé recule vers le fond de l'environnement

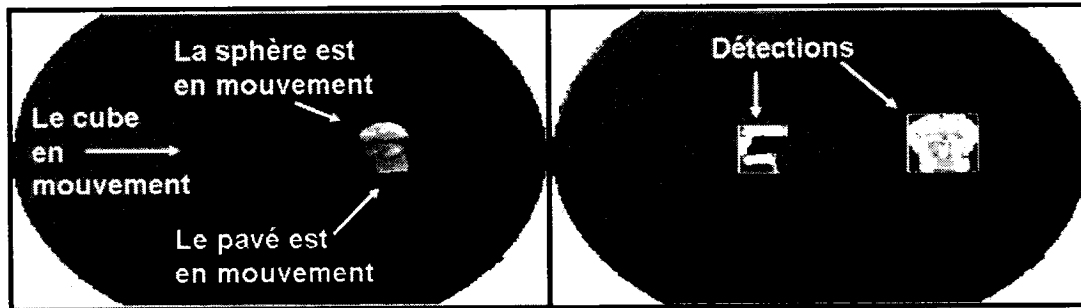


FIG. 4.11 Entre les images 350 et 400 : les trois objets sont en mouvement.

de façon très lente (0.2 m.s^{-1}). La détection est mauvaise et le pavé est encadré de blanc que ponctuellement pendant ce laps de temps. Ce mouvement lent et non propice à des changements caractéristiques sur le plan image est mal détecté.

- Entre les images 150 et 225, il se déplace vers la gauche de la scène (0.24 m.s^{-1}) et est détecté par le système.
- Entre les images 225 et 350, le pavé se déplace vers l'avant et est détecté par intermittence.
- Jusqu'à l'image 400, le pavé regagne sa position initiale en se déplaçant vers la droite (0.21 m.s^{-1}). Il est détecté par le système comme le montre la figure 4.11 dans laquelle il juxte la sphère. Il y a alors occlusion avec la sphère, et le système fait alors l'amalgame entre les deux zones (celle de la sphère et celle du pavé) et n'en présente qu'une seule sur l'écran.

Les mouvements du pavé ont pu illustrer que des mouvements trop lents sont parfois mal détectés par le système.

Il faut noter également que la détection s'accompagne de fausses détections comme par exemple sur la figure 4.10. Cela est dû encore une fois à un mauvais calcul de flux optique qui perturbe le protocole de détection en introduisant de fortes valeurs de flux aux arêtes des objets immobiles et de l'arrière-plan.

L'efficacité de notre protocole a été illustrée dans une séquence d'images virtuelles. Le protocole a été testé sur plusieurs séquences virtuelles dans le but de régler les

seuils de la fusion de région, en particulier pour la taille minimale des zones, et permettre ainsi l'illustration que nous venons de voir.

Il s'agit maintenant de tester les performances du protocole pour une séquence réelle.

4.5.2 Séquences réelles

Le protocole a été testé dans plusieurs environnements différents. Les premiers essais avec des séquences réelles ont montré que le système détecte les objets en mouvement comme l'illustre la figure 4.12. Pour permettre un meilleur suivi des zones en mouvement dans le cas des images réelles, nous avons procédé à deux traitements supplémentaires. Il s'agit tout d'abord d'une érosion pour éliminer les zones trop petites. Ces zones représentent, soit des artefacts de détection, soit des éléments en mouvement trop éloignés de la caméra. Ensuite, une dilatation est réalisée sur les pixels «en mouvement» qui restent, pour réunir des groupes de pixels correspondants au même élément de l'espace 3D, et que la détection aurait séparé. Tous les pixels sont alors réunis en zone par la méthode décrite dans la section 4.4, colorés en blanc et encadrés. Pour effectuer les tests, il a fallu faire évoluer des objets devant le système visuel avec différentes vitesses et à différentes distances de la caméra. Il fut choisi de faire évoluer tout d'abord des êtres humains à différentes vitesses de marche puis une plate forme mobile dont la vitesse pouvait être réglée. Comme un des facteurs restrictifs de la détection est la taille de l'objet dans l'image (la zone doit avoir une taille supérieure à un seuil λ), les piétons et le robot ont évolué à des distances connues du système visuel. Ces distances varient de 1 à 5 m.

Le choix du mouvement de la caméra est un mouvement PAN, car étant donné la disposition du laboratoire, le balayage PAN du système permettait de couvrir de

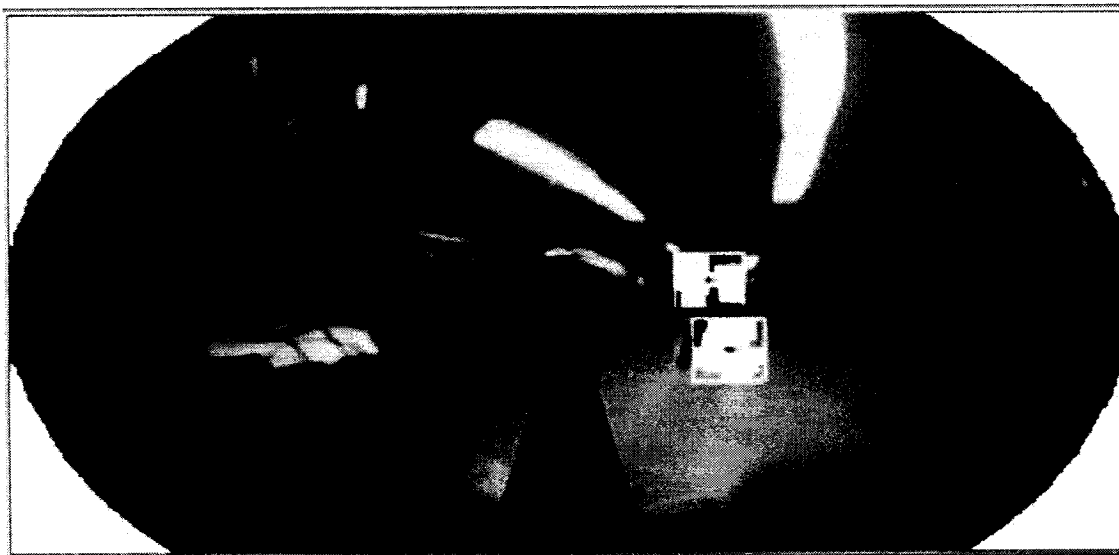


FIG. 4.12 Détection d'une personne en mouvement.

Vitesse	lente	normale	rapide
Distance entre			
1 et 2 mètres	Détection	Détection	Détection
2 et 3 mètres	Détection	Détection	Détection
3 et 4 mètres	Détection intermittente	Détection	Détection
4 et 5 mètres	Détection intermittente	Détection	Détection

TAB. 4.1 Détection de piéton en fonction de la vitesse du piéton et la distance du piéton à la caméra.

grandes zones de l'espace ce facilitait la progression des parties en mouvement pour l'expérimentation, que ce soit des piétons ou des plate-formes mobiles. La vitesse de rotation fut placée à $6,5^{\circ}.s^{-1}$.

4.5.2.1 Personne en déplacement

À défaut de pouvoir utiliser des vitesses précises, trois vitesses de marche furent choisies d'après le sens commun : une vitesse de marche normale, une allure lente et une allure rapide.

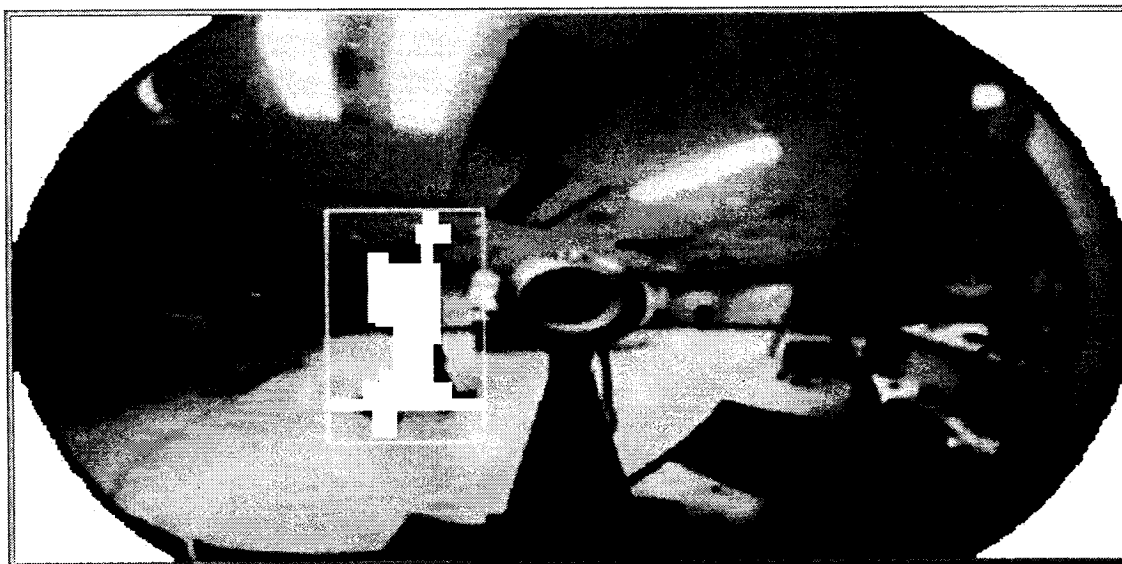


FIG. 4.13 Détection d'un piéton entre 1 et 2 mètres.

Le tableau ci-dessus dresse le bilan des détections obtenues. Comme il était prévisible, le système fonctionne moins bien avec des objets éloignés. La détection est assurée mais non continue. L'objet peut être perdu par l'algorithme de suivi de zone même s'il est souvent retrouvé tant que le piéton est en déplacement. Rappelons que la détection s'effectue à chaque image. Les figures 4.13, 4.14, 4.15 et 4.16 présentent la détection du piéton pour différentes distances. Pour une meilleure présentation, les zones détectées sont représentées uniquement avec le rectangle les entourant. Dans certains cas, le mouvement du corps du piéton peut être ponctuellement décomposé. Les bras et les jambes étant interprétés comme des objets indépendants comme l'illustre la figure 4.17. Ce phénomène provient du fait que la détection a été effectuée pixel par pixel et la réunion des pixels en zones n'est réalisée qu'ensuite.



FIG. 4.14 Détection d'un piéton entre 2 et 3 mètres.

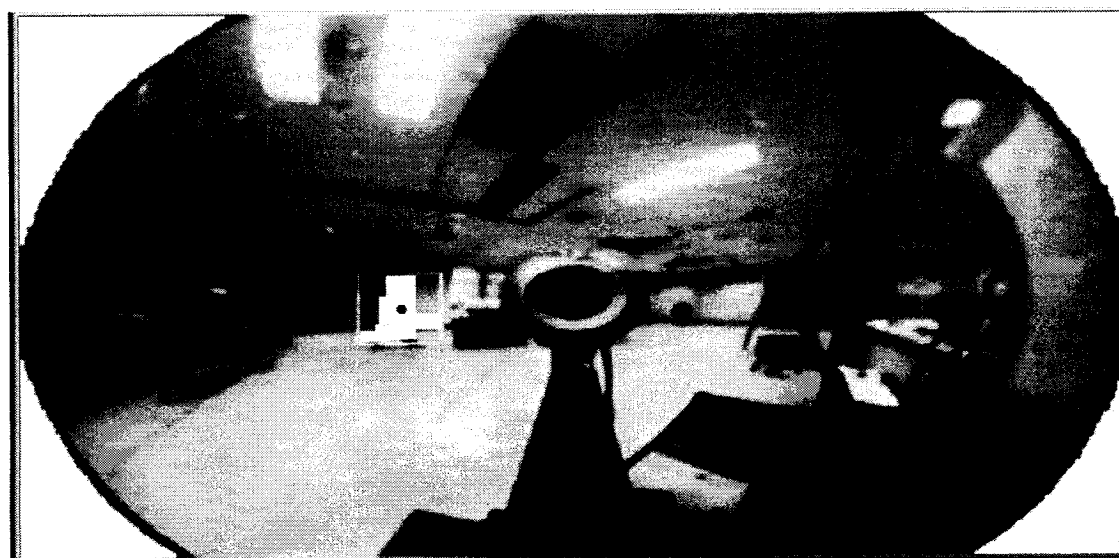


FIG. 4.15 Détection d'un piéton entre 3 et 4 mètres.

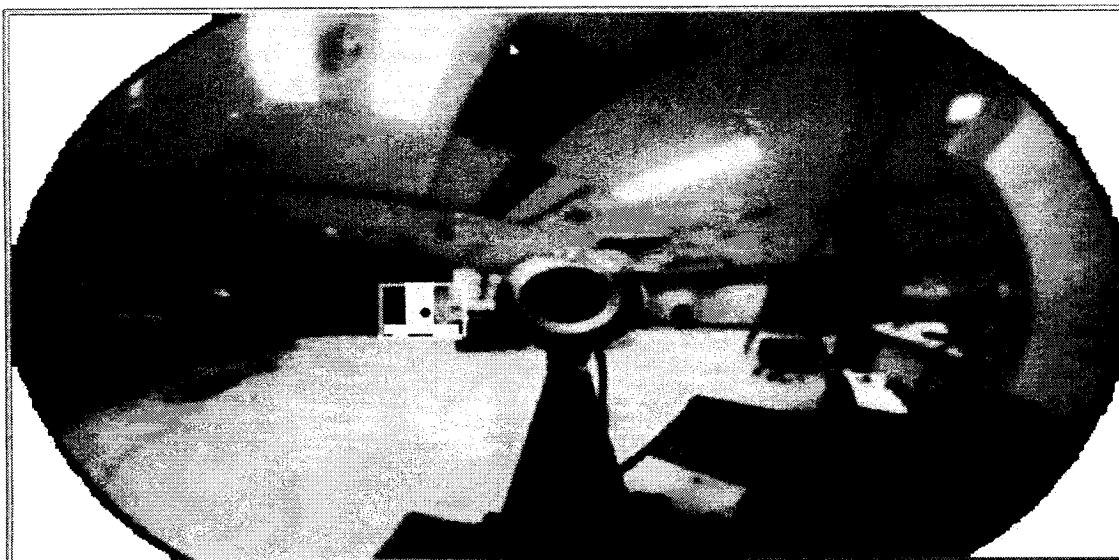


FIG. 4.16 Détection d'un piéton entre 4 et 5 mètres.

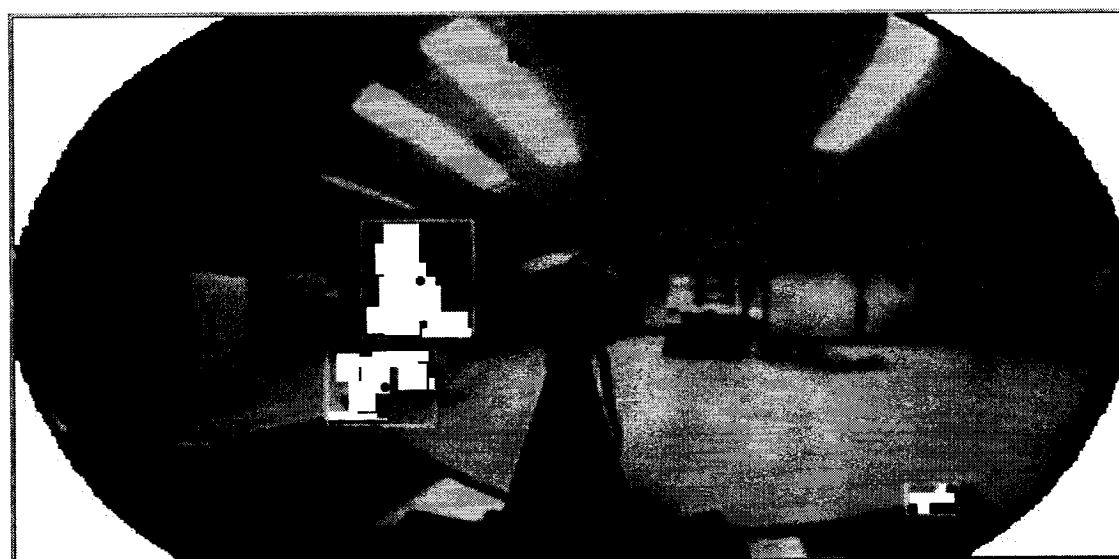


FIG. 4.17 Détection d'un piéton en plusieurs parties avec fausse détection dans l'arrière-plan.

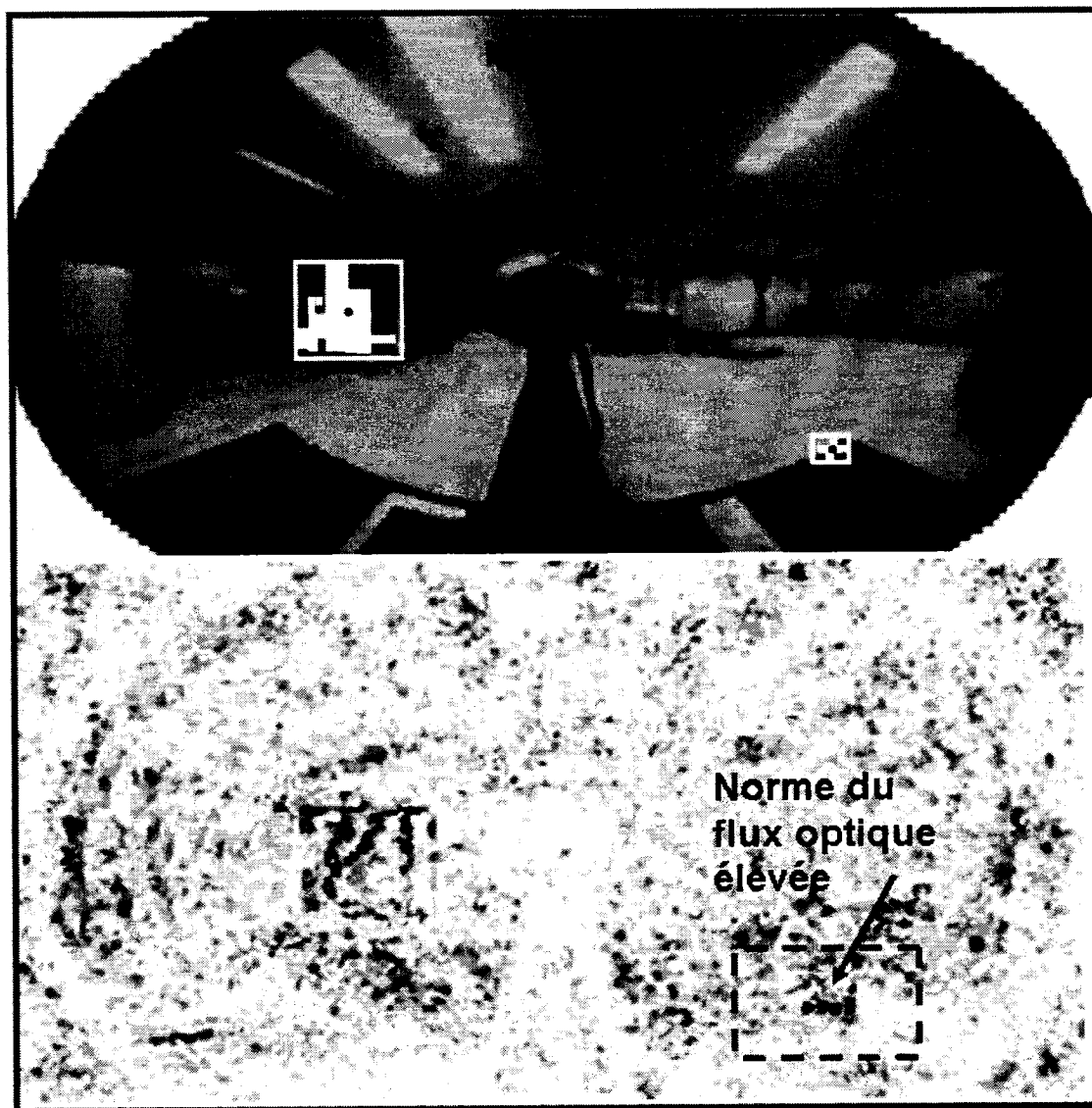


FIG. 4.18 Détection d'un piéton avec fausse détection dans l'arrière-plan. Flux optique correspondant, seuls les pixels où l'amplitude du flux est très importante sont coloriés en noir.

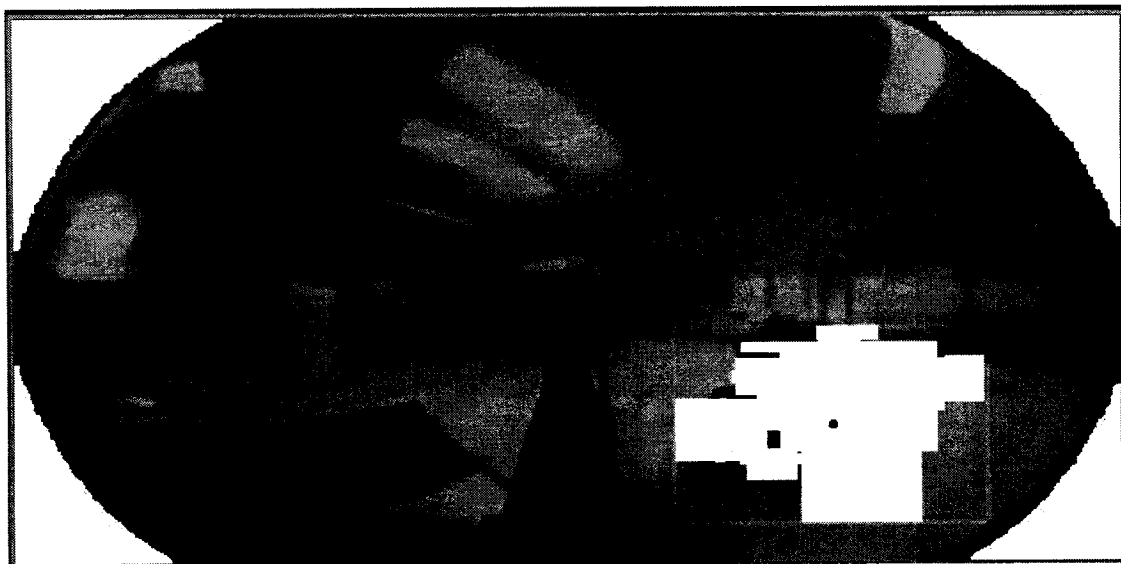


FIG. 4.19 Détection d'un robot en déplacement entre 1 et 2 mètres à 20 cm/s.

4.5.2.2 Plate-forme en déplacement

Le protocole fut également testé avec une plate-forme mobile en déplacement à différentes vitesses allant de 0.2 à 0.5 m.s^{-1} et à différentes distances de la caméra. Pour des distances allant de 1 mètre à 3 mètres le robot est détecté sans problème, comme l'illustrent les figures 4.19 et 4.20. Pour des vitesses supérieures à 40 cm.s^{-1} , le robot est bien détecté pour des distances plus grandes (voir figure 4.21).

La fiabilité de notre système vient d'être illustrée pour des vitesses supérieures à 20 cm.s^{-1} dans plusieurs contextes d'images réelles, celui du déplacement d'un piéton et celui du déplacement d'une plate-forme mobile.

Nous n'avons présenté ici qu'une étude du mouvement PAN, car c'est celle qui fut la plus simple à mettre en place au laboratoire. Le mouvement TILT a lui aussi été testé avec succès et nous pouvons l'illustrer pour le lecteur sur la figure 4.22, dans laquelle une personne se déplace (à gauche) et une autre bouge ses bras (à



FIG. 4.20 Détection d'un robot en déplacement entre 2 et 3 mètres à 50 cm/s.



FIG. 4.21 Détection d'un robot en déplacement entre 4 et 5 mètres à 40 cm/s.

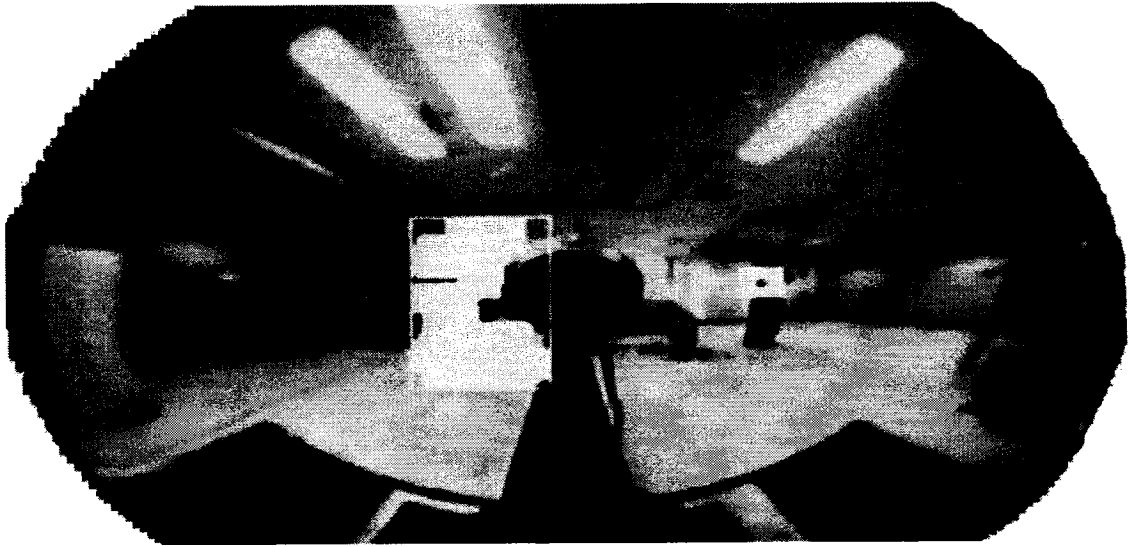


FIG. 4.22 Détection pour le mouvement TILT : deux personnes bougent devant la caméra et sont détectées.

droite).

4.6 Améliorations futures

Le système de détection peut clairement être amélioré.

Par exemple, les zones détectées et marquées en blanc sur les différentes figures, ne reflètent pas exactement l'objet en mouvement qui créent le mouvement. Par exemple, pour une personne se déplacer sous l'éclairage du laboratoire, les ombres changent elles-aussi et créent des variations de flux suffisantes pour être classées comme zones mobiles. Une des voies possibles de recherche est la détection de contours dans l'image et l'assignation des zones «mobiles» à l'intérieur des contours pour refléter vraiment le mouvement d'un objet.

Il a aussi été observé que certaines parties de l'image sont détectées comme mobiles alors qu'elles sont statiques, et cela était dû à un flux optique important dû à une

variation de contraste importante. Une limite supérieure au flux optique pourrait éliminer ces fausses détections.

Comme expliqué précédemment, le seuil de détection des différents critères 4.24 et 4.26 est réglable. Une étude plus approfondie en fonction de l'environnement observé (texture, luminosité, connaissance a priori de la taille des objets en mouvement, etc), pourrait permettre un meilleur réglage du seuil.

Le dernier point est que le protocole se base sur la connaissance des mouvements de la caméra, c'est-à-dire sur les valeurs des vitesses instantanées de rotation des moteurs. L'étape suivante serait un calcul de mouvement propre de la caméra, déjà réalisé souvent dans le cas des caméras en perspective et en cours de perfectionnement pour le cas omnidirectionnel. Si ce calcul est possible et assez rapide, le système n'a besoin que de l'image pour effectuer sa détection.

4.7 Relevé d'informations à plus haute résolution

Si la segmentation basée sur le flux optique reste l'axe principal du système de vision mis en place, la caméra omnidirectionnelle possède le désavantage d'avoir une faible résolution. En plaçant une caméra directionnelle, il est possible de parer à ce défaut.

En premier lieu, si le système comprend un guidage des moteurs pour pouvoir pointer la caméra directionnelle dans la direction souhaitée, il était intéressant de pointer la caméra directionnelle vers des zones en mouvement indépendant détectées par la caméra omnidirectionnelle. Pour cela, la condition suffisante est d'obtenir la direction de la partie de l'espace 3D en mouvement, par rapport à la caméra. Une fois cette direction obtenue, le calcul des angles de rotation PAN et TILT est facile à effectuer.

Comme décrit dans la section 4.4, il est possible de suivre une zone en mouvement

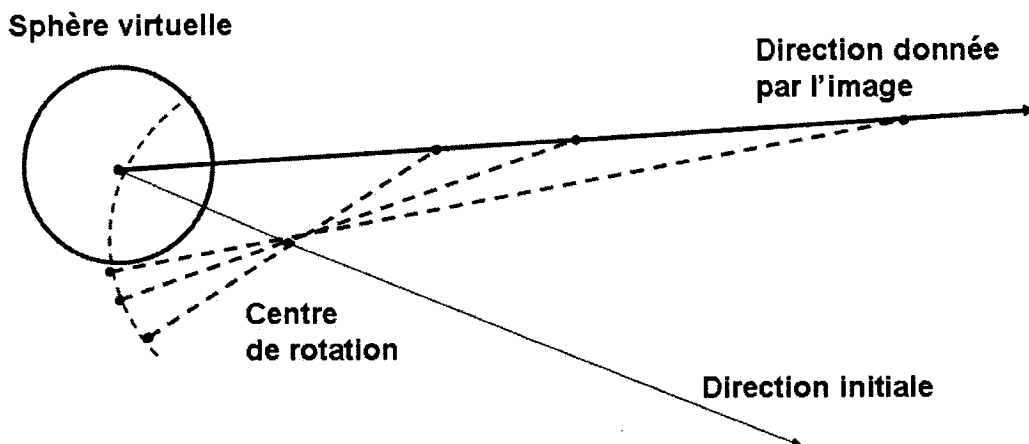


FIG. 4.23 Avec seulement une direction de l'espace, l'objet peut se trouver n'importe où sur la direction.

en mettant à jour entre autres, sa position dans l'image et le nombre NbD d'images successives dans lesquelles elle a été détectée. Si NbD dépasse un certain seuil, que nous avons fixé à 10, nous estimons que la zone en mouvement nécessite une attention particulière et une récolte d'information plus précise.

Le problème qui se pose pour calculer la direction est que même si sa direction spatiale est connue par rapport au centre de la sphère, sa distance au système reste inconnue. Or comme les centres de rotations des mouvements PAN et TILT sont distincts du centre de la sphère, pointer dans la bonne direction devient difficile. Une illustration pour une rotation est donnée sur la figure 4.23. Le même problème va se poser pour l'autre rotation. Une solution consiste à ce que le centre de la sphère appartienne à l'axe de rotation. Etant donné la structure du système, ce ne sera possible qu'avec l'axe de rotation PAN (voir figure 4.4), et représenté sur la figure 4.24.

Dès qu'une zone en mouvement dépasse 10 pour NbD , le barycentre de la zone fournit un point sur la sphère par projection inverse (voir figure 2.3.2) et l'angle

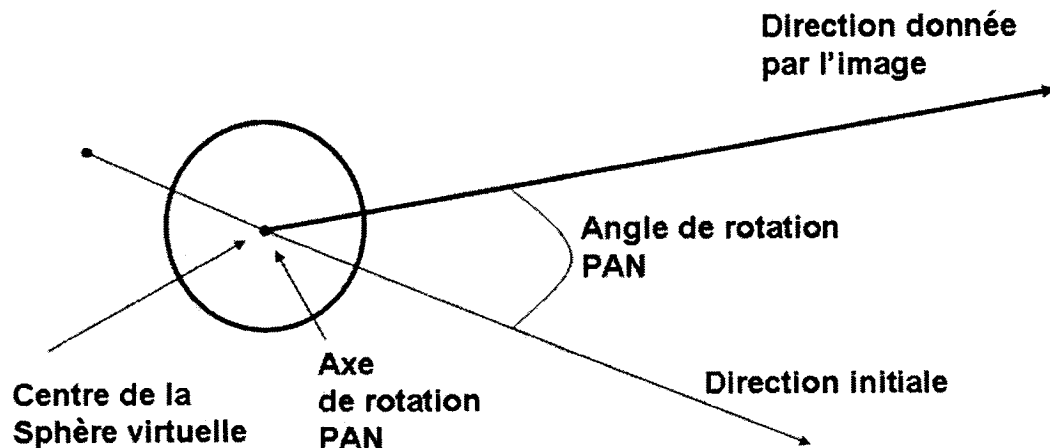


FIG. 4.24 L'axe de rotation PAN passe par le centre de la sphère. Le barycentre de la zone détectée fournit la bonne direction et permet le calcul de l'angle de rotation nécessaire.

de rotation PAN nécessaire est ainsi calculé. Une fois la rotation effectuée à vitesse maximale pour le moteurs (la partie de l'espace en mouvement continue elle aussi peut-être à se déplacer), la caméra directionnelle prend une photo. La figure 4.25 fournit un exemple de détection d'une zone en mouvement ainsi que la photo de la caméra directionnelle. Ce système est déficient puisqu'il ne considère qu'une rotation PAN possible et si l'objet se trouve en hauteur, la photo prise par la caméra directionnelle ne le prendra pas. Néanmoins, étant donné que les caméras utilisées sont des webcams, la plupart des objets susceptibles d'être en mouvement (piétons, plate-forme mobile) n'ont pas de taille significative à plus de 5 m de la caméra omnidirectionnelle. Pour cet intervalle de distance et sachant que la webcam a un champ de vision de 43.2° environ, l'objet détecté se trouve souvent dans le champ de la caméra directionnelle pour nos expériences en laboratoire sachant que les objets en mouvements était du personnel au niveau du sol.

Deux exploitation futures sont à imaginer.

- Tout d'abord le calcul de la position spatiale du point en mouvement en se ba-

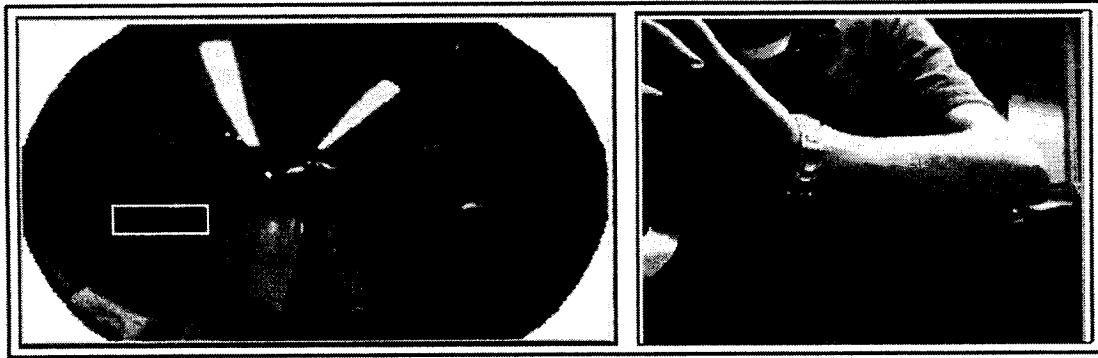


FIG. 4.25 Détection d'une zone en mouvement (bras de l'opérateur) et photo correspondante pour une étude approfondie.

sant sur la position des barycentres dans plusieurs images grâce aux techniques de recouvrement 3D par le mouvement. Là encore, il s'agira de trouver des techniques suffisamment rapide pour ne pas laisser le temps à l'objet de trop se déplacer. Une fois la position trouvée, il est possible de fournir les angles de rotation PAN et TILT nécessaire pour diriger la caméra directionnelle vers la zone en mouvement.

- Ensuite une fois l'objet identifié dans l'image «directionnelle», un système de suivi d'objet peut-être imaginé.

4.8 Conclusion sur le protocole de détection de mouvement

Le protocole de détection de mouvement élaboré durant le projet a été présenté avec ses aspects théoriques et expérimentaux. Les séquences synthétiques dont tous les mouvements étaient connus, et les séquences réelles avec des piétons en déplacement ou des robots en déplacement dans plusieurs environnements ont permis de prouver la fiabilité du système. Afin d'améliorer les performances, certains des seuils utilisés durant la procédure peuvent être adaptée en temps réel durant la détection pour

s'adapter à l'environnement et au type d'objets mobiles qui sont susceptibles d'être filmés dans la zone de l'espace inspectée par le système.

Un des objectifs du mémoire était aussi de pouvoir caractériser la zone en mouvement en terme de vitesse, et de direction. Cette étude n'a pas été faite et mérite d'être dans les développements futurs de ce protocole.

CONCLUSION

La segmentation d'image est l'une des tâches les plus fondamentales de l'analyse d'image. L'une de ses facettes est la détection de mouvement qui consiste à différencier dans une séquence d'images les points objets appartenant à l'arrière-plan, c'est-à-dire l'environnement statique, de ceux appartenant à des objets mobiles. L'utilisation du flux optique apparaît comme un des outils principaux pour cette segmentation.

Le projet avait comme objectif principal la mise en place d'un système visuel alliant vision omnidirectionnelle et vision directionnelle. La vision omnidirectionnelle se fit grâce à un système catadioptrique unissant une caméra et un miroir hyperboloïdal. Les études existantes concernent souvent un autre type de miroir, le paraboloidal. Nous avons donc dû pousser nos calculs dans le cas de notre miroir. Nous avons calculé le Jacobien du passage du plan image vers la sphère et comme cela servait directement notre étude, nous avons calculé les projections de droites de l'espace parallèle au plan image.

Pour mettre en place une segmentation basée sur le mouvement, nous nous sommes naturellement penché sur le calcul du flux optique dans les images omnidirectionnelles. Des techniques déjà existantes dans le cas conventionnel ont été adaptées au cas omnidirectionnel, et en particulier les opérateurs utilisés dans le plan image. Cette étude a permis la mise en place d'un calcul de flux optique par technique différentielle donnant des résultats meilleurs que ceux utilisant des opérateurs homogènes dans toute l'image. Pour tester nos algorithmes de flux, nous avons calculé le flux théorique dans une séquence d'images omnidirectionnelles quand la caméra subit trois mouvements :

- une translation dans l'axe de la caméra,

- une translation perpendiculaire à l’axe de la caméra,
- une rotation autour de l’axe de la caméra.

L’adaptation des opérateurs a été testée sur des séries d’images synthétiques et sur des séries d’images réelles dans le cadre de plusieurs mouvements calibrés.

Pour effectuer une segmentation de mouvement reposant sur le flux optique, nous avons exprimé les modèles de mouvement nécessaires dans le cadre général et dans le cas de mouvements précis fournis par un double moteur PAN et TILT. Les deux caméras posées sur ce double moteur ont permis la validation du protocole de détection dans le cas d’images réelles captées par des webcams. Une autre série de test a été effectuée sur des séquences d’images synthétiques reproduisant la géométrie projective des caméras omnidirectionnelles. La détection est efficace dans le cas réel et dans le cas synthétique. En ce qui concerne la récolte d’informations par la caméra directionnelle, un premier protocole a été mis en place mais doit être amélioré.

Il s’agit aussi de revenir sur les failles de nos développements. Si les résultats obtenus sont souvent meilleurs, les améliorations obtenues dans le calcul de flux ne sont pas toujours significatives dans le cas des images réelles. De plus les algorithmes écrits ne l’ont pas été dans un souci d’optimisation et une étude sur le gain apporté en précision par rapport à la vitesse de calcul pour des applications temps réel serait à réaliser. La lacune la plus importante de notre calcul de flux optique est le manque de test de validation sur l’amplitude du flux optique. Nous n’avons pas disposé pendant le projet de séquences d’images permettant l’analyse de l’amplitude du flux optique. Ce manque de test nous a poussé à ne pas utiliser ce calcul de flux pour notre protocole de détection

L’apport indéniable au protocole de mouvement serait un calcul de mouvement propre de la caméra. Il permettrait, en se servant uniquement de l’image, de cal-

culer le mouvement de la caméra, et de fournir une segmentation. Le calcul du mouvement propre dans une caméra omnidirectionnelle est un sujet étudié par différents groupes de recherche mais à l'heure actuelle, le calcul ne se fait pas en temps réel à cause de la complexité des modèles de mouvement. De plus, son cadre se limite encore à un nombre restreint de mouvements. Pour résumer, les principales contributions scientifiques de ce projet sont :

- le calcul du Jacobien de la projection inverse sur la sphère pour un miroir hyperboloïdal,
- l'expression de l'orientation du flux théorique dans le cas des images omnidirectionnelles pour trois mouvements rigides,
- le développement et le codage d'une technique différentielle de calcul de flux optique adaptée aux images omnidirectionnelles,
- la segmentation basée sur le mouvement dans une caméra omnidirectionnelle,
- la construction d'un système de vision original utilisant toutes les composantes précédentes dans une architecture logicielle avec processus légers et un interface graphique, pour le contrôle des moteurs, le traitement des images et l'affichage des zones détectées.

RÉFÉRENCES

- Barron J.L. et Thacker N.A. (2005). Tutorial : Computing 2D and 3D Optical Flow.
- Makadia A., Geyer C. et Daniilidis K. (2005). Radon-based structure from motion without Correspondences. *Computer Vision Pattern Recognition, 2005*.
- Boyoon J., SukhatmeG.S. (2004). Detecting Moving Objects using a Single Camera on a Mobile Robot in an outdoor Environment. *8th Conference on Intelligent Autonomous Systems*.
- Gandhi T., Trivedi M.M. (2004). Motion Analysis for Event Detection and Tracking with a mobile omnidirectional camera. *Multimedia System, volume 10, n° 2, 2004, pages 131-143*.
- Negishi Y., Miura J. et Shirai Y. (2004). Calibration of Omnidirectional Stereo for Mobile Robots. *Proceedings of 2004 IEEEIRSI International Conference on Intelligent Robots and Systems September 28 - October 2, 2004, Sendai, Japan*.
- Pajdla T. et Micusik B. (2004). Para-catadioptric Camera Auto-calibration from Epipolar Geometry. *ACCV 2004, Korea January 2004*.
- Ying X. et Hu Z. (2004) Catadioptric camera calibration using geometric invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, Issue 10, Oct.2004, Page(s) :1260 -1271*.
- Barreto J.P. et Araujo H. (2003). Paracatadioptric Camera Calibration Using Lines. *IEEE International Conference on Computer Vision, Nice, France - October, 2003*.
- Geyer, C. et Daniilidis K. (2003). Mirrors in motion : epipolar geometry and motion estimation *Ninth IEEE International Conference on Computer Vision, Page(s) :766 - 773, vol.213-16, Oct. 2003*.

- Gandhi T., Trivedi M.M. (2003). Motion Analysis of Omni-Directional Video Streams for a Mobile Sentry. *First AMC International Workshop on video Surveillance*.
- Talukder A., Goldberg S., Matthies L., Ansar A. (2003). Real-Time Detection of Moving Objects in a dynamic Scene from Moving robotic Vehicles. *International Conference on Intelligence Robots and Systems*.
- Vidal R., Shakernia O. et Sastry S. (2003). Formation Control of Nonholonomic Mobile Robots with Omnidirectional Visual Servoing and Motion Segmentation. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on Volume 1, Sept. 2003, Page(s) :584 - 589*.
- Shakernia O., Vidal R. et Sastry S. (2003). Omnidirectional Egomotion From Back-projection Flow, *IEEE Workshop on Omnidirectional Vision (in conjunction with CVPR), June 2003*.
- Stratmann I., Solda E. (2003). Omnidirectional Vision and Inertial Clues for Robot Navigation. *Proceedings of the 11th International Conference on Advanced Robotics (ICAR '03), pages 1915 - 1921*
- Yagi Y., Nishi W., Benson N.E. et Masahiko Yachida (2003). Rolling and Swaying Motion Estimation for a Mobile Robot by Using Omnidirectional Optical Flows. *Mach. Vis. Appl. 14(2) : 112-120 (2003)*.
- Barreto J. et Araújo H. (2002). Geometric Properties of Central Catadioptric Line Images. *Proc. Seventh European Conf. Computer Vision, pp. 237-251, 2002*.
- Daniilidis K, Makadia A. (2002). Image Processing in catadioptric Planes : Spatiotemporal Derivatives and Optical Flow Computation. *Workshop on Omnidirectional Vision (in conjunction with ECCV 2002), pages 3-12, Copenhagen, June 22, 2002*.
- Daniilidis K. et Geyer C. (2002). Para-cata-dioptric Calibration, *IEEE Trans. Pattern Anal. Mach.Intell., 24 :687-695, 2002*.

- Fabrizio J., Tarel J.-P. et Benosman R. (2002). Calibration of panoramic catadioptric sensors made easier. *Omnidirectional Vision, 2002. Proceedings. Third Workshop on 2 June 2002*.
- Frizera Vassallo R., Santos-Victor J. et Jörg Schneebeli H. (2002). A General Approach for Egomotion with Omnidirectional Images, *Proceedings of the Third Workshop on Omnidirectional Vision*.
- Shakernia O., Vidal R. et Sastry S. (2002). Infinitesimal motion estimation from multiple central panoramic views. *Motion and Video Computing, 2002. Proceedings. Workshop on 5-6 Dec. 2002 Page(s) :229 - 234*.
- Stratmann I. (2002). Omnidirectional Imaging and Optical Flow, *Proceedings of the IEEE Workshop on Omnidirectional Vision (OMNIVIS 2002), held in conjunction with ECCV '02 (June 2, 2002, Copenhagen, Denmark), IEEE Computer Society, Los Alamitos, CA, 2002, pp. 104-114*.
- Strum P. (2002). Mixing Catadioptric and Perspective Cameras. *Workshop on Omnidirectional Vision, Copenhagen, Denmark*.
- B.McCane, K.Novins, D.Crannitch et B.Galvin (2001). On Benchmarking Optical Flow. *Computer Vision and Image Understanding 84, 126-143(2001)*.
- Barreto J.P. et Araujo H. (2001).Issues on the Geometry of Central Catadioptric Image Formation, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- Aliaga D. (2001). Accurate Catadioptric Calibration for Real-Time Pose Estimation of Room-Size Environments. *Proc. Eighth International Conference on Computer Vision, vol. I, pages 127-134, 2001*.
- Grossberg M. et Nayar S.K. (2001). A General Imaging Model and a Method for Finding its Parameters. *Proc. Eighth International Conference on Computer Vision, vol. II, pp. 108-115, 2001*.

- Swaminathan R., Grossberg M. et Nayar S.K. (2001). Caustics of Catadioptric Camera. *Proc. Eighth International Conference on Computer Vision, vol. II, pp. 2-9, 2001.*
- Yagi Y., Shouya K., Yachida M. (2000). Environmental Map Generation and Ego-motion Estimation in a Dynamic Environment for an Omnidirectional Image Sensor. *ICRA 2000 :3493-3498.*
- Daniilidis K., Geyer C. (2000). Omnidirectional Vision and Algorithms, *International Conference on Pattern Recognition (ICPR'00)-Volume 1 September 03 - 08, 2000*
- Chang P., Hebert M. (2000). Omnidirectional Structure From Motion. *Proceedings of the 2000 IEEE Workshop on Omnidirectional Vision.*
- Gaspar J., Winters N. et Santos-Victor J. (2000). Vision-Based Navigation and Environmental Representations with an Omni-directional Camera. *IEEE Transactions on robotics and automation, vol 16, No.6.*
- Kang S.B. (2000). Catadioptric self-calibration. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on Volume 1, 13-15 June 2000.*
- Daniilidis K. et Geyer C. (1999). Catadioptric Camera Calibration, *Proc. of IEEE International Conference on Computer Vision, pp. 398-403, Corfu, Greece, 1999.*
- Pajdla T. (1999). Robot Localization Using Panoramic Images, *Proc. Computer Vision Winter Workshop 1999.*
- Baker S., Nayar S.K. (1998). A Theory of catadioptric Image Formation, *Proceedings of the 6th International Conference on Computer Vision Pages 35-42, Bombay, India, January, 1998.*
- Gluckman J. et Nayar S.K. (1998). Ego-Motion and Omnidirectional Cameras In *Proc. Int. Conf. on Computer Vision, pages 999-1005, Bombay, India, Jan. 3-5, 1998.*

- Gluckman J., Nayar S.K et Thoresz K.J. (1998). Real time panoramic stereo. *Proc. of Image Understanding Workshop, IUW98*
- Irani M., Ananda P. (1998). A Unified Approach to Moving Object Detection in 2D and 3D Scences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Pajdla T., Svoboda T. et Hlavac V. (1998). Epipolar Geometry of Central Panoramic catadioptric Cameras, *Lecture Notes in Computer Science*.
- Peixoto P., Batista J., Araújo H. (1998). A Surveillance System Combining Peripheral and Foveated Motion Tracking. *Fourteenth International Conference on Pattern Recognition, 1998*.
- Trucco E. et Verri A. (1998) . Computer vision and applications : A guide for students ans practitioners. *Prentice Hall, March 1998*.
- Nayar S.K. (1996). Omnidirectional Vision, *Proc. of Eight International Symposium on Robotics Research (ISRR), Shonan, Japan, October 1997*.
- Nayar S.K. (1997). Omnidirectional Video Camera, *Proc. of DARPA Image Understanding Workshop, New Orleans, May 1997*.
- Nayar S.K. (1997). Catadioptric Omnidirectional Camera, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Peurto Rico, June 1997*.
- Yamazawa K., Yagi Y., Yachida M. (1997). HyperOmni Vision : Visual Navigation with an Omnidirectional Image Sensor. *Systems and Computers in Japan 28(4) : 36-47*
- Jean-Guy Dion et Rolland Gaudet (1996). Méthodes numériques, de la théorie à l'application. *Collection universitaire de Mathématiques*.
- Barron J.L., Fleet D.J. et Beauchemin S.S. (1994). Performance of Optical Flow Techniques. *International Journal of Computer Vision*.

- Heeger D.J. et Jepson A.D. (1992). Subspace methods for recovering rigid motion. *Int. Journal on Computer Vision*, vol 7, No2, pages 95-117, 1992.
- Simoncelli E.P., Adelson E.H. and Heeger D.J (1991). Probability distributions of optical flow. *IEEE Proc. of CVPR*, pages 310-315.
- Horn B. (1987). Robot Vision. *Cambridge : MIT Press*.
- Curting J.E. (1986). Perception with an eye for motion. *MIT Press, Cambridge, 1986*.
- Bruss A. et Horn B.K.P. (1983). Passive navigation. *In Computer Vision, Graphics, and Image Processing, volume 21, pages 3-20, 1983*.
- Fischler M. A., Bolles R. C. (1981). Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM, Vol 24, pp 381-395, 1981*.
- Horn Berthold K.P. et Schunck Brian G. (1981). Determining Optical Flow. *Artificial Intelligence, 17, pp. 185-203, 1981*.
- Lucas B. et Kanade T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision, *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 674-679.
- Rees D.W. (1970). Panoramic Television Viewing System. *United States Patent (3.505.465)*.

ANNEXE I

RÉSOLUTION DES SYSTÈMES CATADIOPTRIQUES

Si un pixel infinitésimal affiche l'image d'un angle solide infinitésimal $d\nu$ de la scène 3D, et en considérant une aire infinitésimal dA sur le plan image, la résolution de la caméra omnidirectionnelle peut s'écrire :

$$\frac{dA}{d\nu} \quad (\text{I.1})$$

Si nous écrivons la résolution d'une caméra conventionnelle :

$$\frac{dA}{dw} \quad (\text{I.2})$$

Pour la suite du calcul, signalons simplement que les notations peuvent être retrouvées sur la figure I.1. Si nous appelons ψ l'angle que fait la droite joignant le pinhole au centre de l'aire infinitésimale dA avec l'axe optique, l'angle optique sous-tendu par l'aire infinitésimale dA au pinhole est

$$d\omega = \frac{dA \cdot \cos \psi}{\frac{u^2}{\cos \psi^2}} = \frac{dA \cdot \cos \psi^3}{u^2} \quad (\text{I.3})$$

Donc la résolution d'une caméra conventionnelle est alors :

$$\frac{dA}{d\omega} = \frac{u^2}{\cos \psi^3} \quad (\text{I.4})$$

Donc l'aire du miroir que représente l'aire dA est :

$$dS = \frac{d\omega \cdot (c - z)^2}{\cos \psi^2 \cos \phi} = \frac{dA \cdot (c - z)^2 \cdot \cos \psi}{u^2 \cos \phi} \quad (\text{I.5})$$

Avec ϕ l'angle entre la normale au miroir au point (r, z) et la droite joignant le

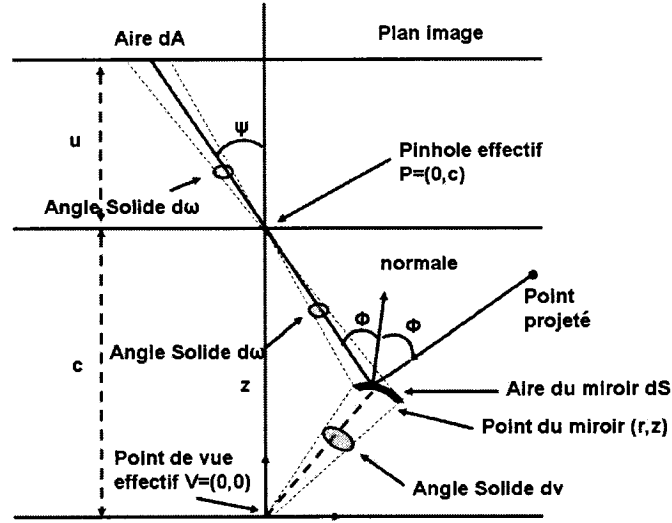


FIG. I.1 Calcul de la résolution d'une caméra omnidirectionnelle.

pinhole et le miroir au point (r, z) . Puisque la réflexion du miroir est spéculaire, l'angle solide pris en image par la caméra catadioptrique est

$$d\nu = \frac{dS \cdot \cos \phi}{r^2 + z^2} = \frac{dA \cdot (c - z)^2 \cdot \cos \psi}{u^2 \cdot (r^2 + z^2)} \quad (\text{I.6})$$

Donc la résolution de la caméra omnidirectionnelle est :

$$\frac{dA}{d\nu} = \frac{u^2 \cdot (r^2 + z^2)}{(c - z)^2 \cdot \cos \psi} = \left[\frac{(r^2 + z^2) \cos \psi^2}{(c - z)^2} \right] \cdot \frac{dA}{d\omega} \quad (\text{I.7})$$

Comme

$$\cos \psi^2 = \frac{(c - z)^2}{(c - z)^2 + r^2} \quad (\text{I.8})$$

nous avons :

$$\frac{dA}{d\nu} = \left[\frac{r^2 + z^2}{(c - z)^2 + r^2} \right] \cdot \frac{dA}{d\omega} \quad (\text{I.9})$$

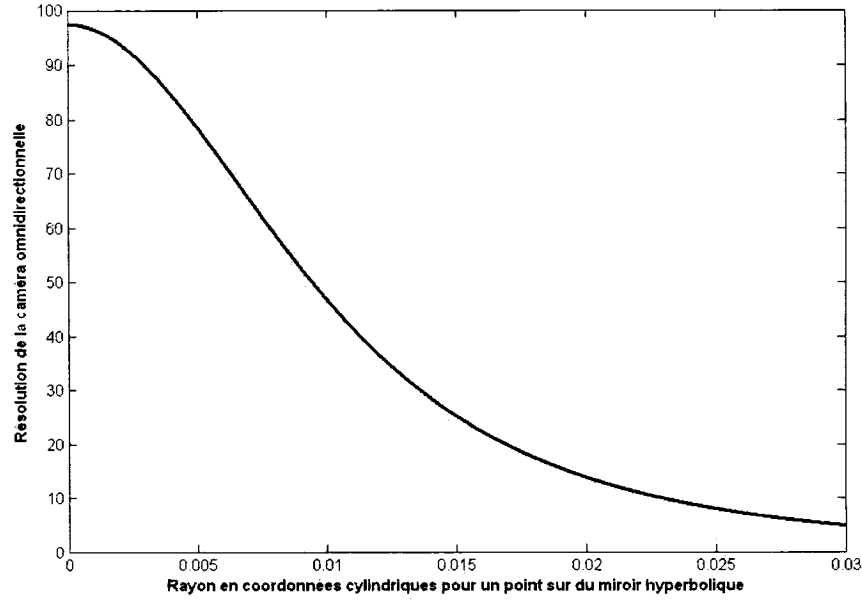


FIG. I.2 Résolution de la caméra omnidirectionnelle avec notre miroir hyperboloïdal.

Ici nous pouvons voir $r^2 + z^2$ qui est le carré de la distance du point sur le miroir au point de vue unique, et $(c - z)^2 + r^2$ le carré de la distance du point au pinhole. Nous pouvons aussi exprimer z en fonction de r :

$$z(r) = c - a\sqrt{1 + \frac{r^2}{b^2}} \quad (\text{I.10})$$

Avec a , b et c paramètres de la caméra.

La fonction :

$$f(r) = \frac{r^2 + z(r)^2}{(c - z(r))^2 + r^2} \quad (\text{I.11})$$

est croissante suivant r . Pour un même angle solide, l'aire qu'il représente sur le plan image est plus petite à la périphérie de l'image qu'au centre. La résolution de la caméra est donc moins grande à la périphérie du miroir qu'au centre. Nous avons tracer figure I.2, l'inverse de la fonction $f(r)$ précédemment exprimée.

ANNEXE II

JACOBIAN DE LA PROJECTION INVERSE

Nous présentons en annexe le Jacobian de la transformation de inverse dans le cas de la parabole c'est-à-dire le cas $l + m < 0$:

$$\frac{\partial \hat{X}}{\partial x} = -x^2 \frac{(1-l^2)}{\sqrt{AB}} - 2x^2 \frac{C}{B^2} + \frac{C}{B} \quad (\text{II.1})$$

$$\frac{\partial \hat{X}}{\partial y} = -xy \frac{(1-l^2)}{\sqrt{AB}} - 2xy \frac{C}{B^2} \quad (\text{II.2})$$

$$\frac{\partial \hat{Y}}{\partial x} = -xy \frac{(1-l^2)}{\sqrt{AB}} - 2xy \frac{C}{B^2} \quad (\text{II.3})$$

$$\frac{\partial \hat{Y}}{\partial y} = -y^2 \frac{(1-l^2)}{\sqrt{AB}} - 2y^2 \frac{C}{B^2} + \frac{C}{B} \quad (\text{II.4})$$

$$\frac{\partial \hat{Z}}{\partial x} = \text{sign}(\hat{Z})x \frac{\{[\frac{(1-l^2)}{\sqrt{A}}] + 2\frac{C}{B}\}(x^2 + y^2) - C\}C}{B^2 \sqrt{1 - \frac{C^2}{B^2}(x^2 + y^2)}} \quad (\text{II.5})$$

$$\frac{\partial \hat{Z}}{\partial y} = \text{sign}(\hat{Z})y \frac{\{[\frac{(1-l^2)}{\sqrt{A}}] + 2\frac{C}{B}\}(x^2 + y^2) - C\}C}{B^2 \sqrt{1 - \frac{C^2}{B^2}(x^2 + y^2)}} \quad (\text{II.6})$$

Avec cette fois-ci :

$$A = (x^2 + y^2)(1-l^2) + (l+m)^2 \quad (\text{II.7})$$

$$B = (x^2 + y^2) + (l+m)^2 \quad (\text{II.8})$$

$$C = l(l+m) - \sqrt{A} \quad (\text{II.9})$$

ANNEXE III

MANUEL D'UTILISATION DU PROGRAMME

L'interface graphique présente une barre de menu avec plusieurs options. Le menu fichier permet de lancer les principales fonctions de base de notre programme.

III.1 Menu Fichier

III.1.1 Lancer les vidéos

Il s'agit dans un premier temps de lancer les vidéos pour afficher simultanément la caméra omnidirectionnelle et la caméra unidirectionnelle. En pressant «Show the videos», il faut pour chaque caméra, choisir le format souhaité (typiquement pour nos caméras 640 par 480 ou 320 par 240), ainsi que les différentes options propres à la Web Cam. Par exemple, l'ajustement du gain la saturation ou le nombre d'images par seconde désirés. Les fenêtres résultantes sont représentées sur les figures III.1 et III.2. Il faut aussi indiquer au système laquelle des deux images est omnidirectionnelle pour lui permettre de bien s'initialiser.

Une fois les vidéos lancées, mise à part la possibilité de choisir un autre format pour l'image, nous pouvons maintenant accéder à plusieurs autres fonctions.

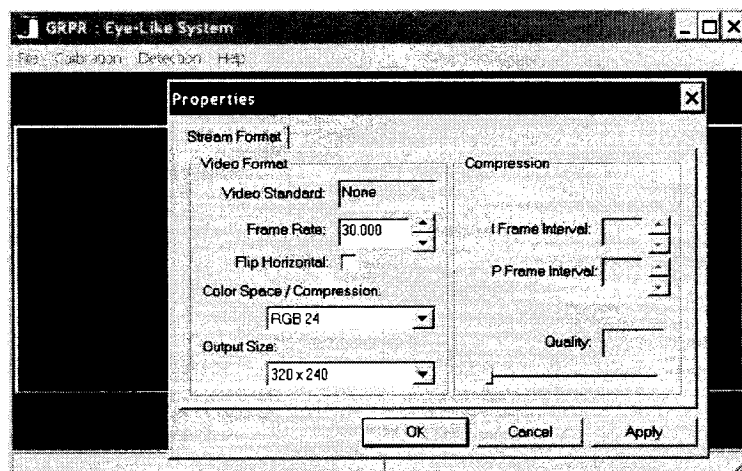


FIG. III.1 Réglages du format et du nombre d'images par seconde

III.1.2 Combinaison des images

Une des possibilités est de recréer le système de vision humaine dont nous avons parlé, en plaçant au centre de l'image omnidirectionnelle, l'image obtenue par la caméra unidirectionnelle. Dans le menu Fichier, l'onglet Combinaison fait apparaître une image combinaison. Un exemple est montré figure III.3. Cette image reste avant tout démonstrative et n'est pas utilisée par la suite mais symbolise l'idée originale de notre projet.

III.1.3 Mouvement des moteurs

Une option permet de diriger à l'aide du clavier les deux moteurs PAN et TILT sur lequel le système est posé. Cette option est située dans le menu Fichier, en utilisant l'onglet «Move the system».

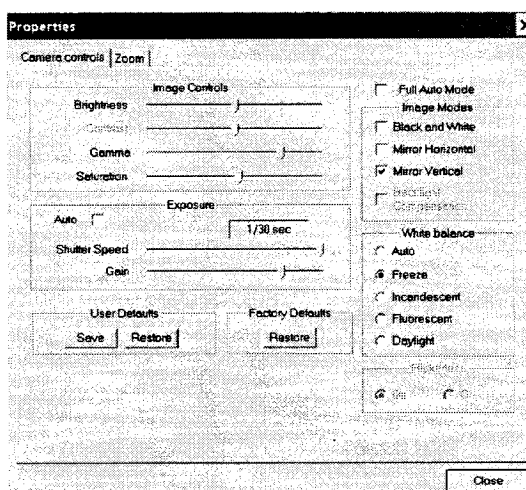


FIG. III.2 Réglages des images.

III.2 Menu Calibration

Il est possible de s'aider à calibrer le système grâce à l'image. L'étude théorique fournit le rayon théorique du cercle de l'image omnidirectionnelle. Le cercle théorique est tracé et en appuyant sur la touche «c», nous obtenons le cercle actuel détecté ainsi que le centre du cercle. Si les deux cercles ne correspondent pas, il faut régler la direction et la position de la caméra par rapport au miroir. Par exemple sur la figure III.4, les deux cercles blancs dessinés ne concordent pas et le système doit être calibrés avec un peu plus de précision.

III.3 Menu Détection

Le dernier menu et non moins essentiel est celui qui lance le thread de la détection de mouvement. En appuyant sur l'onglet «Procedure of Detection», le système s'initialise et lance la détection. Le thread relié à l'affichage des deux images est suspendu et une fenêtre présentant les détection apparaît. Sur cette fenêtre, il est possible de passer du mode «Pixel» qui fournit une détection pixel par pixel en

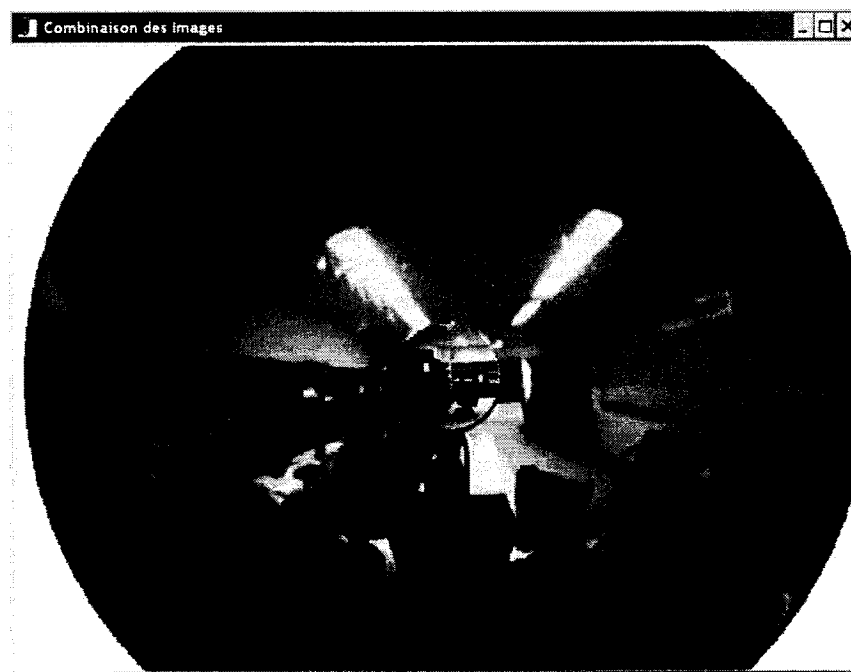


FIG. III.3 Combinaison de deux images.

fonction d'un seuil fixé par l'expérience, au mode «Blob», qui effectue un calcul de probabilité comme décrit précédemment. Les zones sont ainsi réunies en blob, et cela permet d'identifier les zones en mouvement.

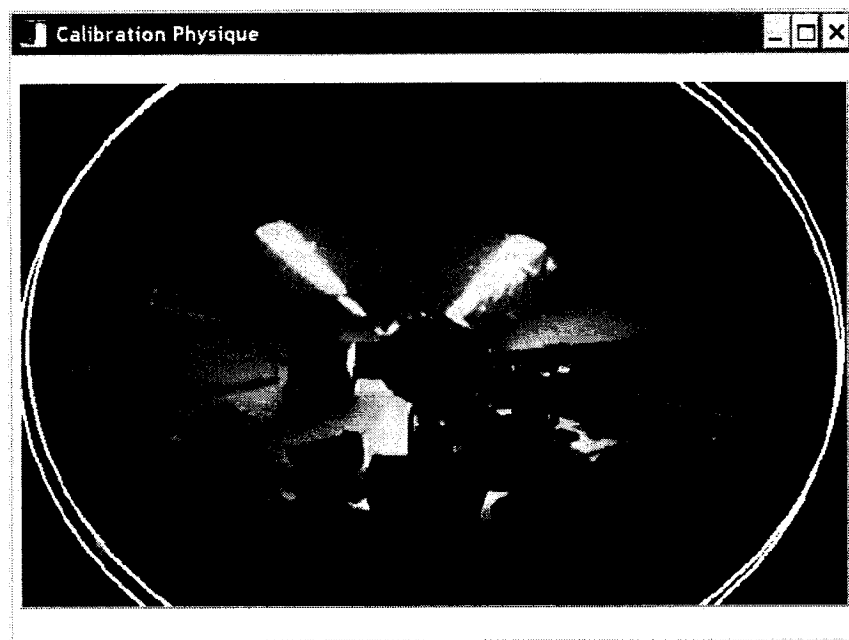


FIG. III.4 Calibration de la caméra omnidirectionnelle.

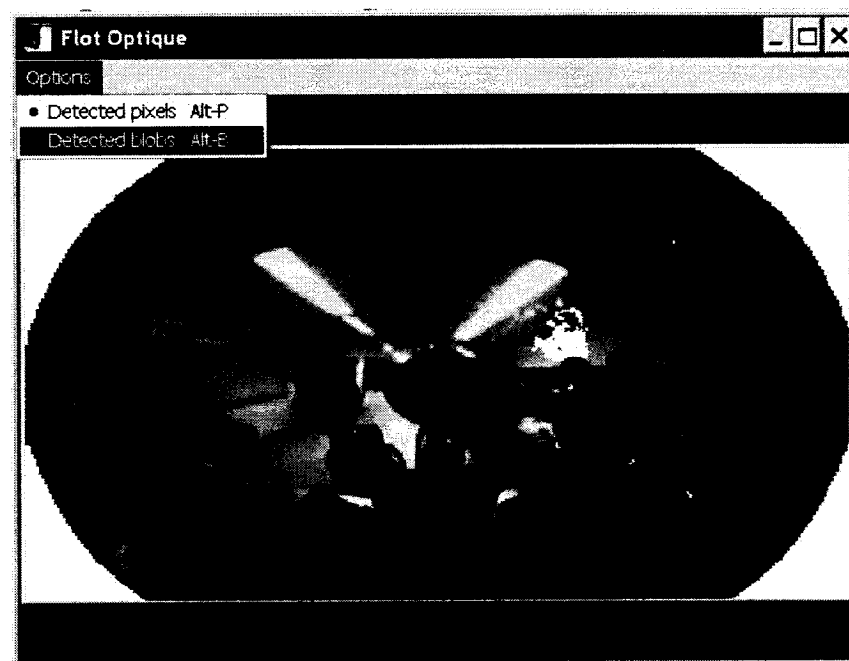


FIG. III.5 Les différents modes de détection.